HELSINKI UNIVERSITY OF TECHNOLOGY
Department of Computer Science and Engineering
Telecommunications Software and Multimedia Laboratory

Kaarle Ritvanen

Protection of Data Confidentiality and Integrity in Radio Communications Systems

Master's Thesis
October 15, 2004

Supervisor: Professor Teemupekka Virtanen
Instructor: Docent Kaisa Nyberg

HELSINKI UNIVERSITY OF TECHNOLOGY	ABSTRACT OF MASTER'S THESIS

Department of Computer Science and Engineering
Degree Program of Computer Science and Engineering

Author:	Kaarle Ritvanen	
Title of thesis:		
Protection of Data Confidentiality and Integrity in Radio Communications Systems		
Date:	October 15, 2004	**Pages:** 89
Professorship:	Telecommunications software	**Code:** T-110
Supervisor:	Professor Teemupekka Virtanen	
Instructor:	Docent Kaisa Nyberg	

Many radio communications systems use cryptographic methods to protect data confidentiality and integrity. In this thesis, those features of a number of systems are studied and analyzed. These systems are GSM, UMTS, WLAN, Bluetooth, IEEE 802.15.3, and IEEE 802.15.4. WLAN security is treated according to the new security enhancements published in July 2004. The analysis presented here reveals security vulnerabilities in some of the systems. In particular, it is shown how key replay and counter rewinding attacks are performed and exploited in GSM, Bluetooth, and IEEE 802.15.3. A few improvements to Bluetooth and IEEE 802.15.3 are also suggested to overcome these problems.

The primary goal of this thesis is to help the reader acquire deeper understanding of the security solutions and their consequences, and facilitate definition of security mechanisms for new systems. Three methods for defining key exchange mechanisms such that they are resistant to replay attacks are given. Moreover, a theory is developed on what parameters should be used to initialize encryption and integrity protection functions. It turns out that there are mainly three classes of initialization parameters. It is also comtemplated how different classes of data should be treated with respect to encryption and authentication.

New high-rate radio systems are being developed, allowing burst transmissions of data frames. However, this imposes challenges to the packet replay prevention mechanism because it turns out that the standard replay counter method is not optimal any more. Efficient solutions for this problem are presented in this thesis, including a new algorithm that does not require the link layer to preserve the order of frames.

Several systems have adopted AES as the cryptographic basis for data protection. Usually, both encryption and integrity protection are desired, and therefore there has been a lot of research concerning authenticated encryption modes of operation for block ciphers. Six such proposals are presented and compared, and their suitability for high-speed communication applications is contemplated.

Keywords:	Authenticated encryption, burst transmission, key establishment, key replay, nonce construction, radio communications
Language:	English

TEKNILLINEN KORKEAKOULU
Tietotekniikan osasto
Tietotekniikan koulutusohjelma

DIPLOMITYÖN TIIVISTELMÄ

Tekijä: Kaarle Ritvanen

Työn nimi:
Tiedon luottamuksellisuuden ja eheyden suojaus radioviestintäjärjestelmissä

Päiväys:	15. lokakuuta 2004	**Sivumäärä:**	89
Professuuri:	Tietoliikenneohjelmistot	**Koodi:**	T-110

Työn valvoja: Professori Teemupekka Virtanen
Työn ohjaaja: Dosentti Kaisa Nyberg

Monissa radioviestintäjärjestelmissä käytetään kryptografisia menetelmiä tiedon luottamuksellisuuden ja eheyden suojaamiseen. Tässä diplomityössä tarkastellaan ja analysoidaan näiden ominaisuuksien toteutusta kuudessa yleisessä järjestelmässä, jotka ovat GSM, UMTS, WLAN, Bluetooth, IEEE 802.15.3 ja IEEE 802.15.4. WLANin turvallisuutta tarkastellaan uusien, heinäkuussa 2004 julkaistujen parannusten mukaan. Tässä esitetty analyysi paljastaa heikkouksia joissakin järjestelmissä, ja siinä näytetään, miten GSM:ää, Bluetoothia sekä IEEE 802.15.3:a vastaan voidaan suorittaa avaimentoisto- ja kellonkääntöhyökkäyksiä. Tässä esitetään myös muutamia parannusehdotuksia näiden ongelmien korjaamiseksi.

Tämän työn ensisijainen tavoite on auttaa lukijaa ymmärtämään paremmin turvaratkaisuja ja niiden seurauksia sekä helpottaa turvamekanismien määritystä uusille järjestelmille. Avaimenvaihtoprotokollan suojaamiseksi toistohyökkäyksiä vastaan esitetään kolme keinoa. Lisäksi kehitetään teoria siitä, mitä parametreja tulisi käyttää salaus- ja eheydensuojausfunktioiden alustamiseen. Osoittautuu, että on pääasiassa kolmentyyppisiä alustusparametreja. Sitä, miten erityyppistä informaatiota tulisi kohdella salauksen ja autentikoinnin suhteen, pohditaan myös.

Nykyään kehitetään uusia korkean tiedonsiirtonopeuden järjestelmiä, jotka mahdollistavat tietokehysten lähetyksen ns. purskeina. Tämä asettaa haasteita toistonestomekanismille, koska tavanomainen, yksinkertaiseen laskuriin perustuva ratkaisu ei tällöin enää toimi optimaalisella tavalla. Myös tähän ongelmaan esitetään ratkaisuja tässä työssä. Yksi näistä on kokonaan uusi algoritmi, joka ei edellytä linkkikerroksen säilyttävän kehysten järjestystä.

Monissa uusissa järjestelmissä tiedon suojaus perustuu AES-salausalgoritmiin. Yleensä halutaan sekä salausta että eheyden suojausta, ja sen vuoksi viime aikoina on tutkittu paljon lohkosalaajien autentikoituja salausmoodeja. Tässä työssä verrataan kuutta tällaista ehdotusta ja pohditaan niiden soveltuvuutta viestintäjärjestelmiin.

Avainsanat: Autentikoitu salaus, avaimentoisto, avaimenvaihto, noncen luonti, purskelähetys, radioviestintä
Kieli: Englanti

Acknowledgements

The opportunity of doing this work has been a very rewarding experience to me. I have learned lots of things concerning cryptology and wireless communications, and I appreciate that very much. First of all, I want to thank my instructor, Docent Kaisa Nyberg, for arranging this great opportunity to do my thesis for Nokia Research Center, and dutifully supporting the writing process.

I am also grateful to Kaisa for encouraging and helping me to publish our ideas in scientific publications. We came up with several new thoughts, when working on various projects related to the area covered by this thesis. It was very interesting to work on real projects, and that must also have contributed advantageously to the content of this work.

I want to thank my supervisor, Professor Teemupekka Virtanen. Moreover, I want to thank my wife Mari for being patient and understanding during the time this thesis was written.

This work was funded by Nokia Research Center and partially by the European Union's Information Society Technology Project 507102.

Contents

List of Figures

List of Tables

Acronyms and Abbreviations

AAD	Additional Authenticated Data
AAL5	ATM Adaptation Layer 5
ACL	Access Control List
ACO	Authenticated Ciphering Offset
AE	Authenticated Encryption
AEAD	AE with Associated Data
AES	Advanced Encryption Standard
AKA	Authentication and Key Agreement
AMPS	Advanced Mobile Phone System
AP	Access Point
ATM	Asynchronous Transfer Mode
AuC	Authentication Centre
AUTN	Authentication Token
AXU	Almost Xor Universal
BCE	Block Cipher Evaluation
BG	Border Gateway
BMC	Broadcast/Multicast Control
BS	Base Station
CBC	Cipher Block Chaining
CBC-MAC	CBC Message Authentication Code
CCM	Counter with CBC-MAC
CCMP	Counter-Mode/CBC-MAC Protocol
CDMA	Code Division Multiple Access
CFN	Connection Frame Number
Chpt.	Chapter
CID	Counter ID
CK	Confidentiality Key
CKSN	Ciphering Key Sequence Number
COF	Ciphering Offset
CRC	Cyclic Redundancy Check
CS	Circuit Switched
CSG	Counter Skew Guard

CTR	Counter
CWC	Carter–Wegman + CTR
DoS	Denial of Service
EAP	Extensible Authentication Protocol
ECB	Electronic Codebook
EU	European Union
GB	Gigabyte(s)
GCM	Galois/Counter Mode
GGSN	Gateway GSN
GLL	Generic Link Layer
GMM/SM	GPRS Mobility Management and Session Management
GMSC	Gateway MSC
GPRS	General Packet Radio Service
GSM	Global System for Mobile Communications
GSN	GPRS Support Node
GTK	Group TK
GTP	GPRS Tunnelling Protocol
HFN	Hyperframe Number
HLR	Home Location Register
HMAC	Hashed Message Authentication Code
IAPM	Integrity-Aware Parallelizable Mode
ID	Identifier
IEEE	Institute of Electrical and Electronics Engineers
IK	Integrity Key
IND	Indistinguishability
IND-CCA	IND under Chosen Ciphertext Attack
IND-CPA	IND under Chosen Plaintext Attack
INT-CTXT	Integrity of Ciphertexts
IP	Internet Protocol
IPR	Intellectual Property Rights
IV	Initialization Vector
kB	kilobyte(s)
kbps	kilobits per second
KCK	Key Confirmation Key
KEK	Key Encryption Key
KO	Key Originator
KSI	Key Set Identifier
L2CAP	Logical Link Control and Adaptation Protocol
LAN	Local Area Network
LFSR	Linear Feedback Shift Register
LMP	Link Manager Protocol
LNK	Link layer
LSB	Least Significant Bit
MAC	Medium Access Control

MAC-A	Message Authentication Code for Authentication
MAC-I	Message Authentication Code for Integrity
MAC-S	Message Authentication Code for Synchronization
MAGNET	My personal Adaptive Global NET
MAP	Mobile Application Part
max.	maximum
MB	Megabyte(s)
Mbps	Megabits per second
MD5	Message Digest 5
MIC	Message Integrity Code
MitM	Man in the Middle
MS	Mobile Station
MSB	Most Significant Bit
MSC	Mobile-services Switching Centre
NMT	Nordic Mobile Telephone System
OCB	Offset Codebook
OMAC	One-Key CBC-MAC
PAN	Personal Area Network
PDCP	Packet Data Convergence Protocol
PDU	Protocol Data Unit
PHY	Physical layer
PIN	Personal Identification Number
PLMN	Public Land Mobile Network
PMK	Pairwise Master Key
PMKID	PMK Identifier
PNC	Piconet Controller
PRP	Pseudo-Random Permutation
PS	Packet Switched
PSK	Pre-Shared Key
PSTN	Public Switched Telephone Network
PTK	Pairwise Transient Key
RAN	Radio Access Network
RANAP	RAN Application Protocol
RC4	Ron's Code #4
RDSR	Replay Detection Shift Register
RLC	Radio Link Control
RNC	Radio Network Controller
RNS	Radio Network System
RRC	Radio Resource Control
RSNA	Robust Security Network Association
SCCP	Signaling Connection Control Part
SDU	Service Data Unit
Sect.	Section
SFC	Secure Frame Counter

SGSN	Serving GSN
SHA-1	Secure Hash Algorithm 1
SIG	Special Interest Group
SIM	Subscriber Identity Module
SPRP	Strong PRP
SS7	Signaling System #7
TDMA	Time Division Multiple Access
TK	Temporal Key
TKIP	Temporal Key Integrity Protocol
UCL	Universal Convergence Layer
UDP	User Datagram Protocol
UE	User Equipment
UMTS	Universal Mobile Telecommunications System
US	United States
USIM	Universal SIM
UTRAN	Universal Terrestrial RAN
VLR	Visitor Location Register
WA	Word Alignment
WCDMA	Wideband CDMA
WEP	Wired Equivalent Privacy
WLAN	Wireless LAN
WPAN	Wireless PAN

Chapter 1

Introduction

According to the Oxford English Dictionary, one meaning of the word *security* refers to safeguarding the interests of an organization or individual against danger, especially espionage or theft [SW89]. Security is usually considered a desirable property of any system. In particular, security is very important to communications systems.

Indeed, when making telephone calls concerning sensitive matters, people do not want anyone else to eavesdrop the conversation. Or when someone uses an Internet bank service to request monetary transfers or issue stock trade commissions, he would probably like that it would not be possible for anyone to modify the account numbers or any other parts of the request on its way to the bank.

Therefore, most countries have laws that make it illegal to eavesdrop phone calls[1] or tamper with financial transactions. However, criminalizing such activities certainly does not prevent performing them but just makes them less attractive, since there is the risk of getting caught and punished.

1.1 Security in Wireless Systems

The wireless communications systems are inherently more insecure than traditional wired systems. Indeed, it is possible to wiretap fixed telephone lines, network cables or routers, but unless he is a dishonest employee of a network operator or telephone company, the eavesdropper must probably break into some physical premises or perform other suspicion-arousing activities. In contrast, with wireless systems the situation is different. As the information is delivered by radio waves, anyone having an appropriate radio receiver is able to retrieve the same information as the intended recipient, assuming he is in the coverage area of the transmission.

This inherent problem of wireless communication is not insurmountable. Cryptographic methods can be used to improve the security of these systems. While legislation only reduces the lucrativeness of wiretapping and interfering

[1] In some countries, such as in Finland, authorities may acquire permission to listen to phone calls of persons suspected of serious crimes [PkL03, Chpt. 5 a, 2 §] although it is not legal under normal circumstances.

with communication between other people, the aim of cryptography is to enforce security by mathematical means [OICT03, Sect. 7.2].

1.2 Security Objectives

There are several desirable features for secure communications systems. Among other things, they include:

Confidentiality Messages (or information) exchanged between communicating parties must remain secret from all unauthorized parties. For example, no illegal eavesdropping of phone calls should be possible. In case of data traffic, collecting passwords or other sensitive data should be impossible.

Authenticity It should be possible to verify that a received message really originated from the party that is claimed to be the sender. If this were not taken care of in the mobile telephone networks, a malicious user impersonating another user could put the cost of his calls on the bill of that user.

Integrity It should not be possible to illicitly alter the contents of a message *en route*. For instance, it should be impossible for any third party to modify requests sent to a banking service.

This thesis concentrates on the three security features listed above. Of course, security is a wide concept and there are many other properties associated with security that must be taken into account when security systems and protocols are designed. For example, availability is one of them. Availability means that no one should be able to prevent other people from using the system. However, this is quite difficult to achieve in wireless communications systems because their operation can be efficiently hindered by a radio jamming device [Stå00], albeit more advanced Denial-of-Service (DoS) attacks make use of the security protocols. On the other hand, there are properties that are even more important in wireless systems than in traditional systems. Anonymity is such an issue. Eavesdroppers should not gain knowledge of whereabouts of any particular user.

1.3 Security in Existing Systems

Many different radio communications systems exist and many of them feature cryptographic security protocols. Security was not a big concern in the design of the first-generation cellular phone networks, such as in the Nordic Mobile Telephone System (NMT) or Advanced Mobile Phone System (AMPS) [Dom02]. In fact, NMT and AMPS phones became subject of commercial cloning, allowing impersonation of other users [WW02, Sect. 15.1]. However, the digital second-generation systems, such as the Global System for Mobile Communications (GSM), employed cryptographic methods to prevent fraud and improve privacy. The security features of GSM and its successor, the Universal Mobile Telecommunications System (UMTS), are discussed in Chapter 2.

Another wireless communication technology that uses cryptography to provide security is the Institute of Electrical and Electronics Engineers (IEEE) standard for Wireless Local Area Networks (WLANs). The security features of the WLAN standard are discussed in Chapter 3. Moreover, the security architecture of the Bluetooth system is presented in Chapter 4. Bluetooth is a standard for short-range wireless communications, or Wireless Personal Area Networks (WPANs) [Blue03a, Part A, Chpt. 1]. Two other WPAN standards, devised by the IEEE, are discussed in Chapter 5.

The six aforementioned systems are interesting because they are either widely used or recently published standards, and therefore chosen to be analyzed in this thesis.

1.4 Problem Statement

New radio communications systems are constantly being developed. To attain success, new standards must address security issues properly and integrate strong cryptographic methods into the system. History has shown that this is not an easy task, since it requires co-operation between system developers and cryptographers. An example of a recent failure is briefly discussed in Chapter 3.

Perhaps the usual approach for designing link layer security features is simply to use the security architecture of another system as a starting point and then adapt it to the system in question. However, better results could be achieved by really understanding the reasons for the solutions, for example, why certain parameters are used in initialization of the keystream generator. One purpose of this thesis is to help the reader gain such understanding, and thus assist in compiling decent security specifications for radio communications systems.

This goal is accomplished by carefully analyzing the link layer security features of existing radio systems listed in Section 1.3. Unfortunately, the specifications of many systems do not reveal the design rationale for these features. Despite the fact, reasons for certain solutions in these systems are surmised in Chapter 6. Some of them are obvious but others are not. In fact, some solutions will turn out to be deficient when they are compared to the corresponding solutions of different systems. Moreover, it will turn out that some systems use too heavy or complex solutions to problems which could have been solved more sensibly. This thesis puts major emphasis on encryption and data authentication[2] methods and the way they are used. Authentication of users or mobile devices is covered to the extent they are related to data protection mechanisms.

Currently, several research projects are concerned with multi-radio access systems. It would be desirable to enable seamless interworking of different access technologies in a single physical device. Section 5.3 discusses attempts to define a generic interface to different underlying radio link layers, and how security features relate to that.

Several new systems use the Advanced Encryption Standard (AES) [NIST01] as the basis for data protection. However, there are several *modes of operation*, that is, ways to use AES (or any other block cipher). Recently, there has been a lot of research concerning Authenticated Encryption (AE) modes, providing both encryption and integrity protection of messages. The differences between

[2] Integrity was also mentioned as a desirable property in Section 1.2. However, terms *data authentication* and *integrity protection* are used somewhat interchangeably in this thesis because the same cryptographic method is used to achieve both of these goals [Sti02, Chpt. 4].

the new AE-type modes of operation, and their properties and suitability for communications systems are discussed in Chapter 7.

1.5 My Contribution

In addition to presenting descriptions of link layer security mechanisms of several wireless communications systems, this work also contributes to the field by several ways. The major achievements are

- showing how Bluetooth encryption is vulnerable to key replay attacks and what kind of consequences it has (Sections 4.3 and 6.3.4),
- summarizing the security-related data flowing between IEEE 802.15 series WPAN link layers and higher protocol layers (Section 5.3),
- devising guidelines for defining new link layer security specifications, including a new replay prevention algorithm and identification of a useful parameter for freshness protection (Chapter 6), and
- comparing features, security, and performance of recently proposed AE modes of operation (Chapter 7).

Chapter 2

Security in UMTS

At the moment, GSM is the most popular system for mobile telephony, accounting for 72 percent of the market [GSMA04]. Therefore, it is reasonable to expect that its successor UMTS will eventually become the dominant third-generation cellular system. UMTS is based on a completely different radio technology, but the core network is still based on the GSM architecture. This chapter describes what kind of security mechanisms are used in UMTS networks. Before dealing with the security issues, it is necessary to present an overview of the UMTS architecture, however. This is done in Section 2.1.

While the primary objective of UMTS was to provide high-rate data services to mobile subscribers, some shortcomings of the GSM security architecture were also addressed when specifying the system. A short comparison with the security properties of GSM is presented in Section 2.5.

2.1 UMTS Architecture

In this section, a short overview of the UMTS network architecture is presented. Also some protocols used between the mobile stations and the network are discussed.

2.1.1 Network

A schematic diagram of the UMTS network architecture is shown in Figure 2.1. The diagram is slightly simplified and does not contain all network components. It represents the organization of a single Public Land Mobile Network (PLMN), a network operated by one operator. The functions of the primary network components are [3GPP04b]:

User Equipment (UE) The physical equipment (usually mobile phone) of the network subscriber, including the Universal Subscriber Identity Module (USIM), which is used to identify him and perform other security-related functions.

Base Station (BS) is responsible for radio transmission between the UE and the RNC [3GPP03d].[1]

[1] *Node B* is the term used for base stations in the specifications.

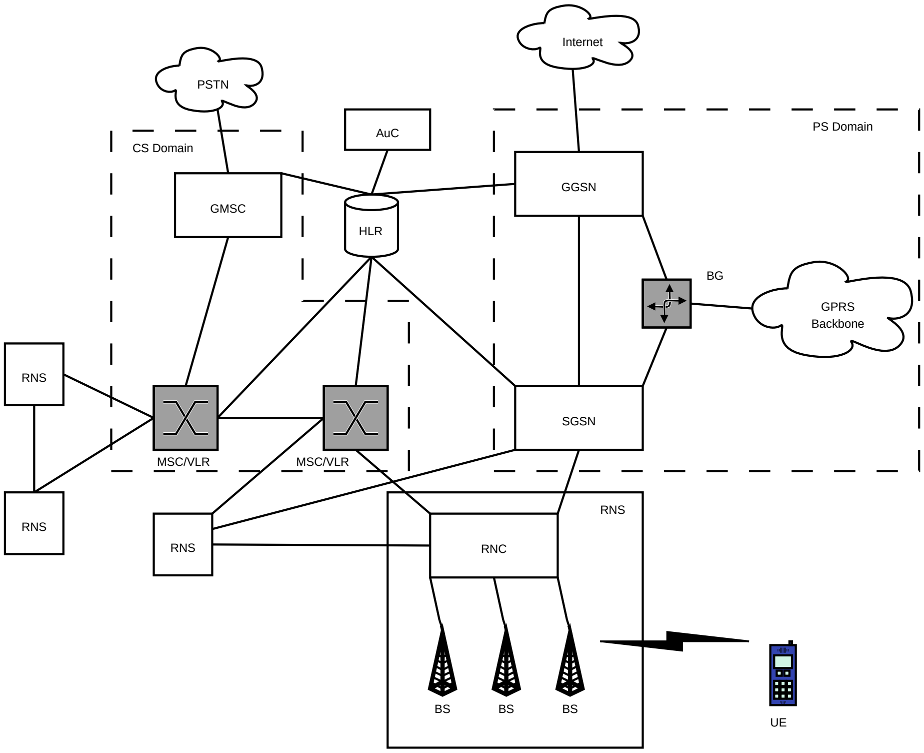

Figure 2.1: UMTS network architecture

Radio Network Controller (RNC) controls a set of BSs and is responsible for controlling the radio resources. RNCs can be interconnected to each other, which allows intra-MSC mobility to be handled at this level. Forms a Radio Network System (RNS) together with the BSs it controls.[2] [3GPP03d]

Home Location Register (HLR) stores important information about the network subscribers. Each user has one HLR entry in an HLR operated by his home operator. One of the most important functions of the HLR is to maintain knowledge about the current locations of the users.

Authentication Centre (AuC) is usually integrated with the HLR. AuC stores user identification keys and generates authentication vectors. AuC is an important component in the security architecture. It is discussed in more detail in Section 2.2.

Mobile-services Switching Centre (MSC) is a telephone exchange which serves subscribers located at a certain geographical area. The task of MSCs is to set up circuit-switched connections so that data can be transferred between desired locations. HLR is queried about the location of the destination when setting up routes, if the destination happens to be another mobile device. Signaling System #7 (SS7) networks are used in signaling between core network components, such as MSCs and HLRs. The major complication of an MSC when compared to an ordinary telephone exchange is the mobility signaling. When a handover occurs so that the participating RNSs are connected to different MSCs, some signaling must take place between the MSCs.

Visitor Location Register (VLR) is typically integrated with an MSC, as depicted in Figure 2.1, but a single VLR could also serve more than one MSCs. VLR stores information about the users that are currently controlled by the MSCs served by it. When the location of a user changes, the VLR responsible for the new location informs the user's HLR about his whereabouts.

Gateway MSC (GMSC) acts as a gateway between the PLMN and an ordinary telephone network (Public Switched Telephone Network, PSTN). Since PSTN exchanges do not know anything about mobility and HLRs, calls from a PSTN are routed to a GMSC, which handles the HLR interrogation and routing to the appropriate location. GMSCs are also needed between two PLMNs unless they have access to each other's HLRs via the SS7 network.

Serving GPRS Support Node (SGSN) is an important component in the Packet Switched (PS) domain. The MSCs transport data in the Circuit Switched (CS) domain, meaning that a constant-rate connection is always established to the destination before data transmission. Nevertheless, UMTS provides also packet switched transmission of data, usually consisting of Internet Protocol (IP) [Pos81a, DH98] datagrams. The

[2] For simplicity, Figure 2.1 depicts only access network components specific to the Universal Terrestrial Radio Access Network (UTRAN). In principle, there might also be legacy GSM Base Station Subsystems parallel to RNSs, for instance.

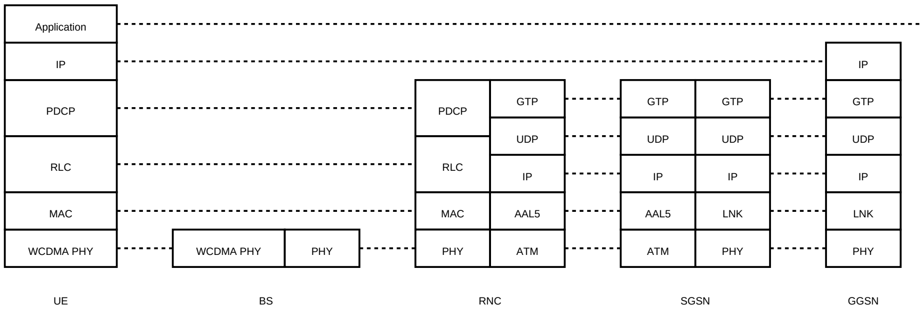

Figure 2.2: User plane protocols in the PS domain

PS domain has evolved from the General Packet Radio Service (GPRS), which is an extension to the GSM network. SGSNs are the equivalent of MSC/VLRs in the PS domain, since they connect to RNSs of a certain geographical location.

Gateway GPRS Support Node (GGSN) acts as a gateway to a packet data network, which typically is the Internet [3GPP04c].

Border Gateway (BG) acts as a gateway to the inter-PLMN GPRS backbone. If a user is roaming in another PLMN, his packet traffic is tunneled from the local SGSN to the GGSN of his home network through the backbone. [3GPP04c]

2.1.2 Protocols

Figure 2.2 shows the usual user plane protocols of the PS domain. The UTRAN physical layer between the UE and BS is based on Wideband Code Division Multiple Access (WCDMA) radio technology [HT04]. The link layer is divided into sublayers, namely Medium Access Control (MAC), Radio Link Control (RLC), and Packet Data Convergence Protocol (PDCP). Although not shown in Figure 2.2, there is still one sublayer, named Broadcast/Multicast Control (BMC), which is used when transmitting the same content to several UEs [3GPP04g]. [3GPP03a]

The MAC layer maps *logical channels* to *transport channels*, which are in turn mapped to *physical channels* by the physical layer [3GPP04d, 3GPP04e]. Different logical channels carry different types of data, and several such channels may be mapped onto the same transport channel. The MAC layer takes care of multiplexing and priority handling of those different flows. MAC layer also handles UE identification on channels common to all UEs. [3GPP04f]

The RLC layer handles segmentation of frames to smaller units suitable for transmission. RLC operates in three different modes:

Transparent mode Service Data Units (SDUs) are transmitted as they are. No headers are prepended to the frames. Only segmentation and reassembly take place.

Unacknowledged mode Detection of missing Protocol Data Units (PDUs) due to transmission errors.

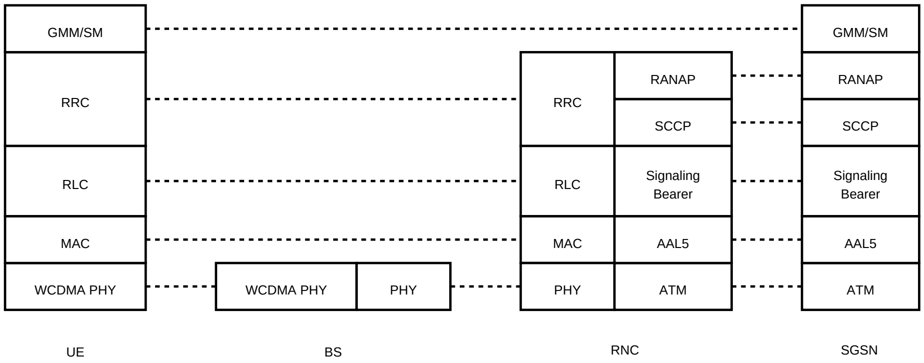

Figure 2.3: Control plane protocols in the PS domain

Acknowledged mode Automatic retransmission of missing PDUs and detection of duplicate PDUs.

When operating in unacknowledged or acknowledged mode, ciphering is applied at the RLC level. In transparent mode, ciphering takes place at the MAC level [3GPP04f]. [3GPP03b]

The primary purpose of PDCP is to compress the headers of the upper layer protocols, such as IP, Transmission Control Protocol [Pos81b], User Datagram Protocol (UDP) [Pos80], and Real-time Transport Protocol [SCFJ03], which are likely to be used in the PS domain, thus saving valuable radio link bandwidth. [3GPP03c]

Figure 2.3 presents the control plane protocols of the PS domain [3GPP03a]. The Radio Resource Control (RRC) protocol is used to carry control information over the radio link. The RRC protocol routes mobility and connection management control messages submitted by higher layers, and establishes control connections and user plane radio bearers,[3] among other things. [3GPP04h]

2.2 Mutual Authentication and Key Exchange

It is obvious that the network has to verify the identity of the users before allowing them to initiate or receive calls, or to use any other services. Otherwise it would be possible to make calls and put the cost on someone else's account, or to intercept calls or short messages intended to others, for instance.

On the other hand, similarly as the network needs protection against malicious users, also the user needs protection against certain attacks. An active attacker operating a fake RNS might try to set up the same security context several times by a replay attack. Or it might try to collect large amounts of authentication challenge–response pairs from the USIM, in order to solve the secret identification key.[4] UMTS uses mutual authentication to provide protection against attacks of this kind [3GPP03e].

[3] Term *bearer* refers to information transmission paths that can be created and deleted, except for the signaling bearers used by RRC itself. In contrast, *channels* are predefined entities that exist at physical and MAC layers. They cannot be created nor destroyed. [3GPP04a, 3GPP03a]

[4] This kind of attack is possible in the GSM system but would probably not be very feasible, however. Assuming the most common authentication algorithm COMP128, it would take

The key components in mutual authentication are the UE, the currently serving MSC/VLR or SGSN, and the HLR/AuC of the user's home network. MSC/VLR and SGSN play the same role in the authentication procedure, depending on the domain for which the authentication is carried out. User authentication as well as mobility management are performed independently for both domains [3GPP04b].

In UMTS, terminal authentication and session key exchange are combined into a single procedure, which is called the Authentication and Key Agreement (AKA) procedure. The procedure is based on the fact that a secret user identification value (K) is stored in both AuC and the user's USIM. MSC/VLR or SGSN wishing to initiate authentication retrieves a set of authentication vectors from the HLR/AuC using the Mobile Application Part (MAP) protocol [3GPP04i]. One such vector is used immediately and the rest are stored for future authentications. Unused authentication vectors can be further passed to another VLR when the location of the UE changes [3GPP04b]. Authentication vectors consist of the following parts:

Challenge which is sent to the UE.

Authentication Token (AUTN) which is also sent to the UE. AUTN contains a message authentication code that UE uses to verify that the challenge was really generated in the correct AuC. AUTN also contains a sequence number to protect the UE against replay attacks.

Response which is the correct answer to the challenge (also depending on K).

Confidentiality Key (CK) that is used to encrypt transmissions.

Integrity Key (IK) that is used to authenticate control messages.

Authenticating MSC or SGSN sends the challenge and AUTN to the UE, which in turn verifies the AUTN, calculates the response from the K stored in the USIM and the challenge, and sends the response back. The UE's response is verified after that by the network. [3GPP03e]

In addition to calculating the response, different one-way functions are used to derive CK and IK from the challenge and the secret key. Of course, this happens both in USIM and in AuC so that CK and IK are never transmitted over the radio link. After a successful AKA exchange, CK and IK are known to the UE and the MSC/VLR or SGSN. The MSC or SGSN communicates the keys to the RNC by using MAP, and they are used as session keys to protect subsequent signaling and the sessions established after the authentication. As authentication is performed independently for both domains, there are different session keys for both domains too. [3GPP03e]

Further information about mutual authentication and key agreement can be found in [NN03], for example.

2.3 Encryption

As mentioned in Section 2.1.2, encryption can be done at either MAC or RLC layer, which means that encryption is limited to happen only between the UE

several hours even using a smartcard reader [WGB98], not to mention an over-the-air attack.

and the RNC. Encryption is only applied to RLC SDUs. This is true also when encryption layer is MAC because then RLC operates in transparent mode and no RLC headers are prepended. In acknowledged mode, RLC entities exchange some flow control messages but they are not encrypted. [3GPP03b]

Figure 2.4 shows how encryption is applied. The encryption algorithm is a proprietary stream cipher called *f8*, which is a special mode of operation built around the KASUMI block cipher [3GPP02b]. In addition to CK, it takes a few more inputs. The length parameter specifies how long the keystream shall be, the maximum allowed length for the keystream being 20,000 bits.[5] The direction and bearer identity are taken into account to avoid reusing the same keystream for both directions or on different radio bearers. The counter in turn ensures that the same keystream is never used twice on the same bearer for a single direction.[6] [3GPP03e, 3GPP02a]

The counter parameter is maintained per bearer basis and it consists of two parts: a short sequence number and a longer Hyperframe Number (HFN). When ciphering takes place at the RLC layer, the sequence number is equal to the RLC frame sequence number used to detect transmission errors. The number is transmitted in cleartext in the headers of RLC PDUs. In acknowledged mode, its length is 12 bits, whereas in unacknowledged mode it is 7 bits. In transparent mode, the 8-bit Connection Frame Number (CFN) maintained by the MAC layer is used as the sequence number.[7] Since the total length of the counter is always 32 bits, the length of HFN varies accordingly. [3GPP03b, 3GPP03e]

HFN is incremented every time the shorter counter wraps around. Whenever a new bearer is created, the 20 most significant bits (MSBs) of its HFN are initialized to a value (START) stored in the USIM. When switching the RNC, it is therefore possible to omit authentication if it has previously been done[8] but still avoid reusing keystreams, even if bearer identifiers (IDs) are reused. The value of START is always updated before it is used according to the maximum HFN of all bearers. When a new AKA exchange takes place, START is set to zero because the keys are changed. [3GPP03e]

Although the network usually initiates the AKA procedure, the lifetime of session keys is not solely determined by the network. USIMs contain a parameter named THRESHOLD that is the maximum allowed value for START parameter. If it is exceeded, new AKA must be initiated. [3GPP03e]

[5] In a known plaintext attack on a stream cipher, the attacker easily acquires knowledge of the corresponding keystream bits. The length of the keystream is sometimes limited to prevent the attacker from gaining too many consecutive keystream bits, in which case he might be able to deduce the subsequent bits or even the encryption key. However, this limitation is not due to the properties of the *f8* encryption algorithm. It is just a requirement arising from the fact that the maximum length of physical layer frames is 20,000 bits [3GPP04j].

[6] Reusing the keystream reveals partial information about the plaintext to an eavesdropper. See [NN03, Sect. 2.1.3.1] for an illustrative example.

[7] In principle, CFN is incremented after each frame. However, if several frames are transmitted during the same physical layer timeslot, CFN of the first frame is used and the keystream generator is not reinitialized at the frame boundaries. [3GPP04f]

[8] AKA can be omitted because the session keys can be communicated within a VLR or SGSN area using the MAP protocol. Each pair of CK and IK is associated a value named Key Set Identifier (KSI). On handover, the UE sends the KSI of the keys currently stored in the USIM to the new RNC, in order to avoid possible inconsistencies between the currently active keys on both sides.

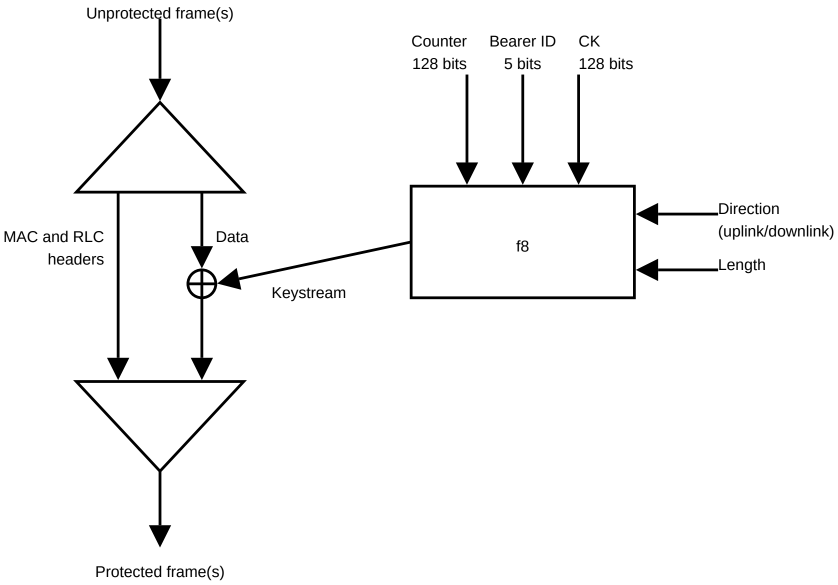

Figure 2.4: Link layer encryption in UMTS

Data class	Encryption	Authentication
User data	Yes	No
RLC control data	No	No
RRC	Yes	Yes

Table 2.1: Data encryption and authentication in UMTS

2.4 Control Message Authentication

UMTS employs message authentication and integrity checking at the RRC layer [3GPP04h]. Therefore, authentication is not applied to user data nor RLC flow control messages, since RRC runs above RLC. As this is the case, RRC messages can also be encrypted, in addition to authentication. Table 2.1 summarizes how encryption and authentication are applied to different classes of data.

Authentication uses a special function named *f9*, which is another mode of operation around the KASUMI algorithm [3GPP02a]. Inputs to *f9* include, in addition to IK and the message to be authenticated, a message sequence number and a transmission direction bit, just like in encryption, though their purpose is different. In encryption, the purpose of these inputs is to prevent keystream reuse, but in authentication, they are used to prevent message replay. The sequence number is comprised of HFN and a shorter sequence number included in each RRC frame. HFN initialization depends on the same START value as in encryption. [3GPP03e]

Moreover, *f9* requires yet another input parameter, namely a random value generated by the RNC. This parameter is called FRESH, and it is transmitted in the beginning of the RRC connection and used throughout that connection.

It prevents control message replays by the UE (or an attacker pretending to be one). In other words, the FRESH parameter protects the network from malicious terminals, whereas the sequence number protects users from false networks, since the START value is determined by the USIM. [3GPP03e]

In contrast to *f8*, there is no radio bearer identity input to *f9*, due to a historical reason [NN03, Sect. 2.1.4]. As there are several signaling bearers, this may seem to facilitate replaying RRC messages on different signaling bearers. However, this problem has been fixed by attaching the bearer identity to the message to be authenticated [3GPP04h].

Not all RRC messages are integrity protected [3GPP04h], for obvious reasons. For example, it is not possible to apply protection before FRESH and START have been transmitted and IK has been derived.

If a message is to be protected, *f9* outputs a 32-bit Message Authentication Code (MAC-I[9]) [3GPP02a]. This value is inserted to the message. When the message is received, the receiver recomputes the value from the message contents and checks whether it matches the MAC-I. Figure 2.5 shows how integrity protection works in RRC. [3GPP04h]

2.5 Comparison with GSM

The GSM security architecture naturally has many features similar to those of the UMTS network. The main differences are:

- GSM does not support network authentication nor prevent key replay. This leaves room for an active attack that is described in Section 2.5.2.

- GSM control messages are not integrity protected, which is a major security problem. See [WW02, Sect. 15.9.5] for examples how this shortcoming can be exploited. All error checking and correction codes are linear [ETSI00d], so encrypting them with an additive stream cipher does not contribute much to authenticity.

- In GSM, encryption is applied at the lowest possible level, even lower than channel coding [ETSI00b].[10] Therefore, in a known plaintext attack, the attacker gains knowledge of even more consecutive keystream bits than he knows those of the plaintext. Even ciphertext only key recovery attacks are possible.

2.5.1 Encryption in GSM

Figure 2.6 shows how encryption works in GSM. There is a ciphering key (Kc) derived from the identification key (Ki) and the authentication challenge, just like in UMTS. Kc together with a physical layer frame sequence number determines the keystream, which is split into two parts, one used for the downlink and the other for the uplink stream. The sequence number is bound to the clock

[9] The message authentication code is denoted by MAC-I to avoid ambiguities with the MAC layer, and also to distinguish it from another authentication values called MAC-A and MAC-S.

[10] In fact, there are two information bits per burst that are not encrypted. These bits are called *stealing flags* and they indicate whether control data are passed instead of user data along the data stream. [ETSI00d]

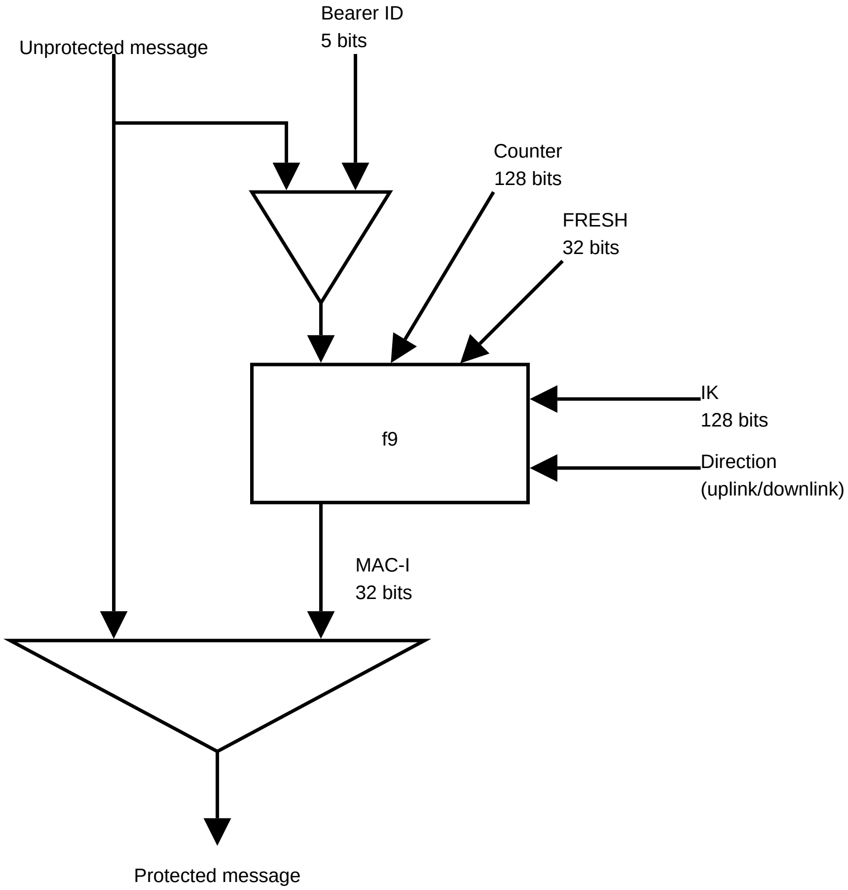

Figure 2.5: Integrity protection in RRC

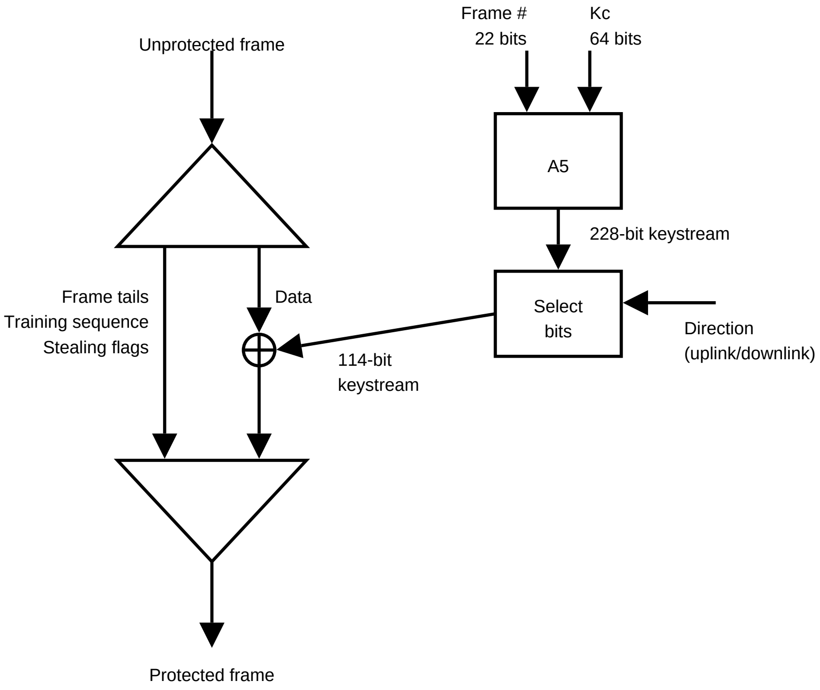

Figure 2.6: GSM ciphering

of the base station and thus known to the Mobile Station (MS) due to synchronization. The sequence number wraps around every 3.5 hours. [ETSI99b]

Originally, the encryption algorithms used to protect over-the-air GSM traffic, called A5/1 and A5/2, were kept secret as an additional security measure [MP92, Sect. 4.3.3]. However, they were completely reverse engineered in 1999 [BGW99]. Several attacks on them have been published. The weaker version, A5/2, is already completely broken, as the session key can be recovered from a few dozen milliseconds of ciphered conversation [BBK03]. Pretty efficient time–memory tradeoffs exist against A5/1, which allow key recovery in a few seconds or minutes, depending on the amount of available ciphertext. These attacks require a huge amount of preprocessing and disk storage. See [BSW01] for details.

A5/2 was introduced later than A5/1 because certain organizations opposed using strong cryptography in confidentiality protection. However, as data integrity in GSM depends solely on encryption, integrity protection of data was weakened too. Nowadays, when online A5/2 key recovery is possible, an active Man-in-the-Middle (MitM) impersonator can hijack telephone calls. The attacker listens to the victim call, and once he has collected an adequate amount of ciphertext and recovered the key, he closes the session with the victim user and takes over the call.

2.5.2 Key Replay Attack

Elad Barkan, Eli Biham, and Nathan Keller presented an interesting active attack on the GSM network. The attack leverages the lack of network authentication and key replay prevention mechanism, and was originally presented in [BBK03, Sect. 7.1].

The idea is that an attacker records an A5/1-encrypted session, including the authentication challenge, which together with the (unknown) Ki uniquely determines the encryption key. Later, the attacker impersonates the network to the user and sends the recorded challenge to the user, which causes the same encryption key to be used within this session. As the network is free to choose any encryption algorithm the client supports [ETSI99b], the attacker chooses A5/2,[11] obtains an adequate amount of ciphertext, and recovers the key. After that, he can use the key to decrypt the previously recorded conversation.

[11] This is always possible because it is mandatory to implement A5/2 [ETSI99a]. However, removal of this requirement has been proposed due to this attack [Bro04].

Chapter 3

WLAN Security

IEEE 802.11 is the standard defining how WLANs should operate. It contains definitions for both physical layer and link layer, the latter of which is also called the MAC layer. The original version of the standard [IEEE97] indeed addressed security issues, but there were serious flaws in the security specification, however. For example, using linear checksums encrypted with an additive stream cipher[1] does not provide very good integrity protection. Moreover, stream cipher initialization vectors (IVs) could have been made longer than 24 bits. The vulnerabilities of this original security protocol, Wired Equivalent Privacy (WEP), and potential ways to exploit them are discussed in [BGW01].

The IEEE has recently updated the security specification to strengthen the security of WLANs [IEEE04]. The updated standard defines how WLAN stations can establish Robust Security Network Associations[2] (RSNAs) between each other. An RSNA between stations allows them to authenticate each other using various methods, establish fresh cryptographic keys, and use effective algorithms for data protection. This chapter describes this improved security scheme because it is more relevant if we want to examine mechanisms that could be reused in future communications systems.

There are two operational modes for WLANs:

***Ad hoc* mode** where the stations communicate directly with each other, and

Infrastructured mode where there is a fixed Access Point (AP) that meditates all the traffic between other stations and possibly provides further connectivity to other networks and APs. Using this mode indeed requires that some infrastructure is installed to the area where the stations are operated.

The operational mode used affects the security mechanisms. In *ad hoc* mode, RSNAs are established with each other station to be communicated with. In contrast, if an AP is present, an RSNA is typically formed only with it.

An overview of station authentication methods is given in Section 3.1. After that, data protection protocols are presented and discussed. The description of the WLAN architecture presented here is based on [IEEE99] and [IEEE04].

[1] The cipher was RC4, a proprietary stream cipher by RSA Security, Inc. However, the details of the algorithm had already been leaked to the public [Sch96, Chpt. 17].

[2] The security associations of this new type are called robust because those of WEP were apparently not.

3.1 Station Authentication

WLAN station authentication is based on the generic port-based access control model defined for Local Area Networks (LANs) [IEEE01]. RSNA establishment between two stations may begin with an Extensible Authentication Protocol (EAP) [ABV$^+$04] exchange. EAP is a generic framework for performing authentications, possibly using a distinct authentication server. The authentication methods used with EAP produce a shared secret between the parties, which is called the Pairwise Master Key (PMK). The PMK is a long-term link key that is used to generate session keys for data encryption and authentication.

The EAP specification defines authentication protocols for one-time passwords and token cards. A mechanism for MD5-based [Riv92] authentication analogous to the Challenge Handshake Authentication Protocol [Sim96] is defined too. Additional protocols are currently being specified. One such protocol is EAP-AKA, which allows authentication using USIMs and the UMTS AKA protocol [AH04]. EAP-SIM is a similar protocol for GSM Subscriber Identity Modules (SIMs) [HS04].

Nevertheless, it is not mandatory to use EAP. It is also possible to use a pre-shared key (PSK) directly as the PMK. This is possible because the possession of the correct PMK is verified during subsequent steps of the RSNA establishment procedure.

Each PMK has an identifier, which is derived from the station addresses and the key itself using HMAC-SHA1-128 [KBC97, NIST95]. This 128-bit PMK Identifier (PMKID) is unique with an overwhelming probability, given any reasonable number of PMKs. It is useful when a station has been absent from the network and wants to join it again. If both parties have cached the key, it is not necessary to run the authentication procedure again. The associating[3] station just has to send the PMKID. The identifiers are also useful when the link key can be communicated between several APs because then authentication can be omitted when stations switch between those APs.

3.2 The 4-Way Handshake

A procedure called *4-way handshake* completes RSNA establishment. In accordance with the name, four messages are exchanged between the stations, as shown in Figure 3.1. The supplicant is the station that authenticates itself to the authenticator. If an AP is involved, it sends the first handshake message because APs always act as authenticators when establishing RSNAs.

The main function of the 4-way handshake is to generate short-term session keys to be used with data protection algorithms. Both parties generate random numbers[4] and exchange them. These values alongside with the parties' MAC addresses[5] and the PMK are used as inputs to a key derivation algorithm pro-

[3] In WLAN terminology, *association* just means establishing a connection with another station. In infrastructured mode, normal stations initiate associations only with the AP.

[4] These random numbers are called *nonces* by the specification, as shown in Figure 3.1. However, in this thesis, we follow the terminology of [WHF03], where *nonce* refers to any value, random or not, that is not repeated in the same context.

[5] MAC address is a unique 48-bit number identifying the device. Each frame must contain the MAC addresses of the sender and the receiver, since the frames are transmitted on a shared physical channel.

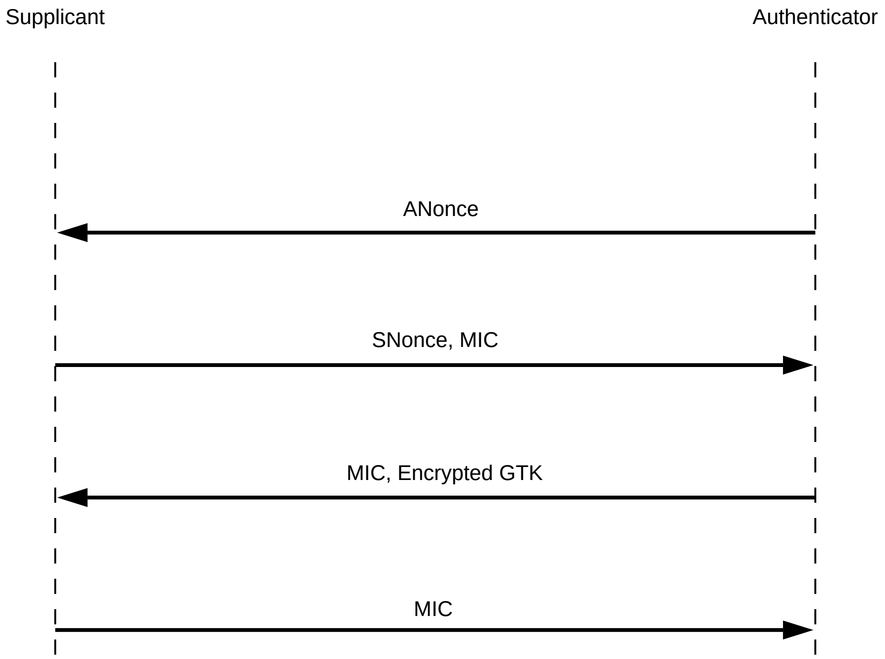

Figure 3.1: The 4-way handshake

ducing a Pairwise Transient Key (PTK) as a result. The supplicant computes the PTK after the first step and the authenticator after the second step. The key derivation algorithm is based on HMAC-SHA-1 where the PMK is used as the key.

The PTK is divided into three parts:

Key Confirmation Key (KCK) that is used to compute and verify the Message Integrity Codes[6] (MICs) in the handshake messages. The authentication algorithm is either HMAC-MD5 or HMAC-SHA1-128.

Key Encryption Key (KEK) that is used to encrypt the Group Temporal Key (GTK) included in the third message. The encryption algorithm is either RC4 or AES.

Temporal Key (TK) that is used with the encryption and data authentication algorithm to protect unicast traffic.

The MICs are calculated over the handshake messages using KCK. They verify that the parties share a common PMK and make it possible to use PSK authentication, in addition to preventing active attacks on the 4-way handshake.

TK is used to protect unicast traffic between the stations, but there is a different key for multicast and broadcast traffic, namely GTK. This key is essentially a random number chosen by the authenticator and is used to protect non-unicast traffic originating from it. TK can be renewed by re-executing the

[6] The WLAN specification uses rather term *Message Integrity Code* than Message Authentication Code to avoid ambiguities with acronyms.

4-way handshake, but GTK can be changed by a lighter group key handshake using the old KCK and KEK.

It is good to note that in infrastructured mode, only the AP needs to have a GTK, since the other stations never have to broadcast anything as they are communicating with the others via the AP. However, in *ad hoc* mode, the 4-way handshake must be performed twice, so that both parties send their own GTK, if the both parties wish to send non-unicast messages. In principle, it is possible to execute the 4-way handshake once and then use the group key handshake for the other direction. However, this is not mandated to allow for simple implementations.

3.3 Data Encryption and Authentication

In WLANs, both encryption and integrity protection are applied to user data. The invariable MAC headers of data frames are also integrity protected, but all control messages are left unprotected. In contrast, some UMTS control messages are even more protected than user data, as said in Section 2.4. Possible reasons for this are investigated in Section 6.5.

The enhanced WLAN standard defines two modes for data protection:

- Counter-Mode/CBC-MAC Protocol (CCMP)
- Temporal Key Integrity Protocol (TKIP)

The standard allows using also vendor-specific algorithms. Stations advertise the cipher suites they support, and the station willing to associate chooses one of them and indicates its choice in the association request. This happens before the 4-way handshake and possible EAP exchange.

Although TKIP is defined by the standard, it is not required to implement it. TKIP is a workaround that tries to fix the most serious shortcomings of WEP by turning the IV to a 48-bit counter and adding a non-linear integrity checking code. It is designed so that it would be possible to upgrade legacy devices to use it. However, this implies that the integrity checking code cannot be very strong. Otherwise the legacy devices might not be able to compute it online.

CCMP is more interesting because it is mandatory to implement in devices claiming RSNA compliance and is currently believed to provide good security. CCMP uses AES in Counter with CBC-MAC (CCM) mode [WHF03], thus providing both encryption and data authentication. Figure 3.2 shows how CCMP works. The CCM mode itself is discussed in more detail in Chapter 7 and Appendix A.

The nonce required by the CCM mode is constructed from a 48-bit sequence number, a MAC header field, A2, and the frame priority.[7] A2 contains the address of the transmitter, so it makes the keystream different for both directions even though the same sequence number and priority were used. The sequence number must be incremented for each packet, thus preventing keystream

[7] The frame priority is a 4-bit value affecting the CCM nonce construction. However, it is not transmitted along the frame. The specification says that the priority is always set to zero and reserved for future use.

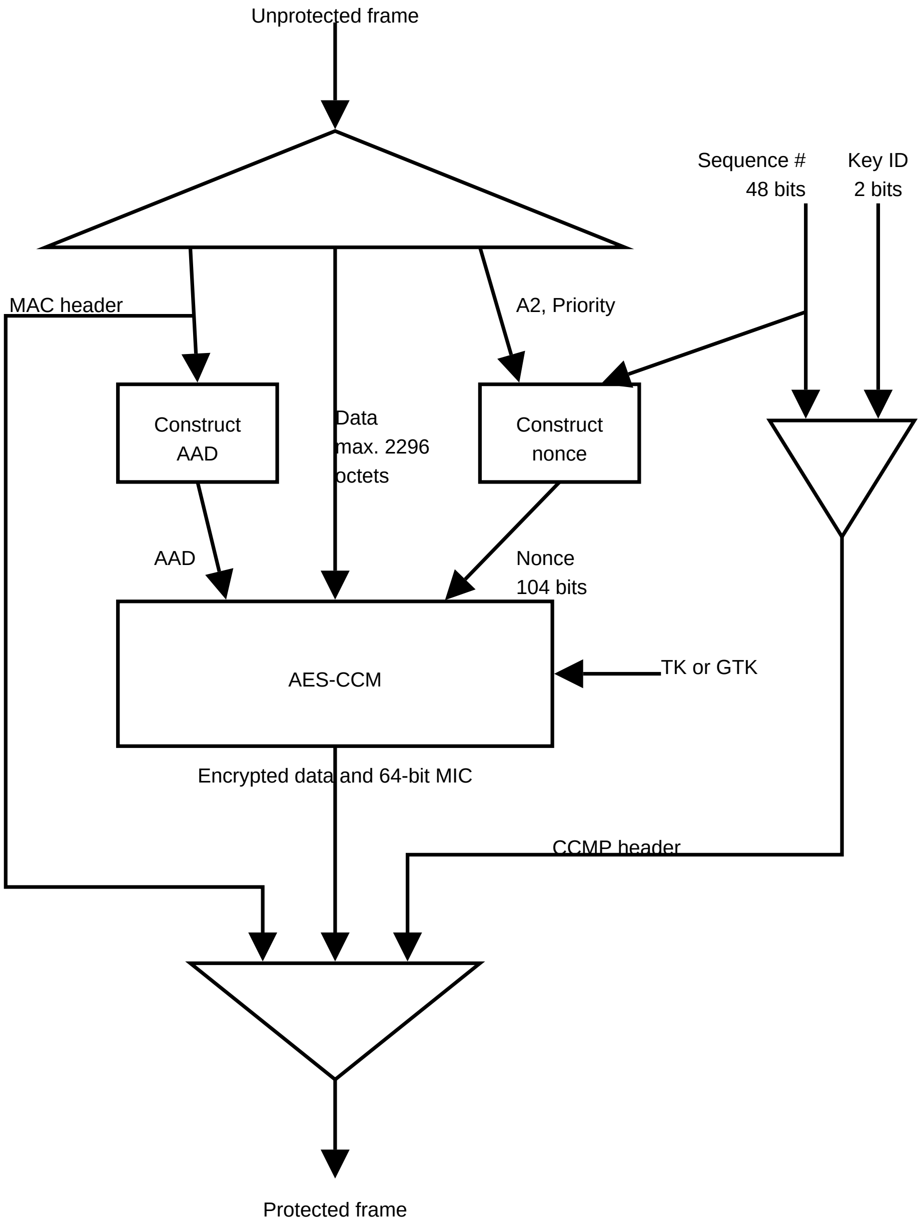

Figure 3.2: CCMP operation

reuse and enabling detection of replayed frames. With 48 bits counter wrap-around does not happen very frequently. Nevertheless, the specification mandates changing the session key if wrap-around happens because it states that no sequence number should be used more than once.

The sequence numbers are maintained separately for each TK and GTK. As TK associations are bidirectional, there must be different sequence number registers for transmission and reception. Transmission counters are incremented when frames are sent, the values being transmitted along them, whereas reception counters are used to detect frame replay. Interestingly, the specification says that stations must maintain separate reception counters for each priority. The reason is that then it is possible to reorder frames according to their priorities *en route*,[8] without confusing the replay prevention system. The priority classes can be thought to form separate logical channels that are multiplexed to a single transport channel.

[8] Unlike many other radio communications systems, in infrastructured mode of WLAN it is possible to reorder frames *en route*, since they are routed via the AP.

Chapter 4

Bluetooth Security

Note: Parts of this chapter will be separately published in [RN04].

Bluetooth is a wireless communications standard intended to be used in WPANs, being developed by an industry consortium named Bluetooth Special Interest Group (SIG). Many handheld devices, such as mobile phones, personal digital assistants and laptop computers incorporate a Bluetooth radio to enable low-cost wireless data transmission. This chapter presents the Bluetooth security mechanisms and also points out a few problems they have. The descriptions of the security features are based on [Blue03b]. Two other WPAN specifications are considered in Chapter 5.

Typically, WPAN technologies require that there is one *master device* that controls the network. The other devices, called *slaves*, synchronize their clocks to that of the master and adjust their radio transmission and reception to that. The network formed of devices synchronized in this way is called a *piconet*.

4.1 Authentication and Key Establishment

This section discusses authentication and session key establishment procedures in Bluetooth. Moreover, two related concepts, namely *pairing* and *link keys* are presented.

4.1.1 Pairing

Forming a security association between two Bluetooth devices is called pairing. In practice, pairing involves entering a common Personal Identification Number (PIN) to the devices. Or if either of the devices does not have a suitable user interface, it may have a fixed PIN that must be entered to the other device. The crucial assumption is that the potential attacker does not see the PIN that is entered to the devices. The PIN together with a publicly exchanged random number is used to compute an *initialization key* that is used as a *link key* for a while.

4.1.2 Link Keys

Link key is a shared secret between the communicating devices. In principle, there are four types of link keys:

- combination keys
- unit keys
- temporary keys[1]
- initialization keys

Unit keys are used by devices with limited memory resources. In practice, it means that the device with limited resources uses the same link key with all other devices. This implies that using them is very insecure, since it facilitates mutual eavesdropping among different communicating peers. In fact, their use is deprecated and they are ignored in this discussion.

Temporary keys are used in point-to-multipoint configurations. As the name suggests, such configurations are usually relatively short-lived. Applications may make the slaves use a common encryption key derived from this common temporary link key to allow encryption of broadcast traffic. Note that also unicast traffic is encrypted with the common key when a temporary link key has been set up, as will be explained in Section 4.2.1. After the master has finished broadcasting that needs encryption, the slaves can be told to fall back to the previous link keys.

In most cases, the link key is a combination key, denoted by K_{AB} in the specification. The key is derived from the addresses of the devices and random numbers generated by both of them, hence the name combination key. According to the specification, combination keys are *semi-permanent*, in the sense that they can be changed but typically have long lifetimes. In fact, the specification suggests that combination keys can be stored into non-volatile memory and used to authenticate and generate encryption keys for future sessions. So it is reasonable to assume that link keys do not change very often in point-to-point configurations.

In addition to these four types of link keys, there is an interface for importing link keys. Therefore, it is possible to use higher layer key exchange protocols, for instance. These imported keys have the same function as unit and combination keys although not specifically listed in [Blue03b, Part H, Sect. 3.1].

Whenever the link key is changed, the related message exchange is protected by the previous link key. Initialization keys are the first link keys and used only for this purpose just after pairing, and they are discarded after a new link key has been generated. One could argue that changing the link key does not provide any additional security, since if he sees all the messages and knows the previous link key,[2] an attacker can easily derive the new key. This is true, but if he misses even a single message related to changing the link key, the attacker

[1] In [Blue03b], keys of this type are called *master keys* and denoted by K_{master}, but this term is a bit misleading. In specifications of some other wireless communications systems, such as those of WLANs [IEEE04], long-term link keys (combination key equivalents) are called master keys.

[2] This assumption is realistic and can be accomplished by an exhaustive PIN search, for example. This was pointed out in [JW01, Sect. 4]. Although the paper discusses Bluetooth version 1.0B, the attack can be applied also to the newest version with a small modification.

is unable to derive the new link key or any subsequent link keys generated in future. Although not explicitly stated in the specification, it seems that this is the rationale of the link key changing procedure, to provide security in practice, if not in theory.

A change of the link key is followed by two other procedures, which are

1. Mutual authentication
2. Encryption key exchange (if encryption is used)

Authentication and key exchange are allowed to happen at any other time too, but they are mandatory after link key renewal. Section 4.1.3 explains how authentication works in Bluetooth and Section 4.1.4 shows how encryption keys are agreed on.

4.1.3 Authentication

Bluetooth uses a special one-way algorithm named E_1 when authenticating other devices. It is based on the SAFER+ block cipher [MKK98]. The inputs to E_1 are:

- current link key
- device address of the claimant[3]
- 128-bit challenge

The challenge is generated by the verifier and sent to the claimant. Both parties run E_1 and the claimant sends the response to the verifier that checks whether the results match. E_1 produces also another result, which is called Authenticated Ciphering Offset (ACO). This 96-bit value is used in key exchange and is discussed in Section 4.1.4.

Authentication always takes place for both directions after the link key has been changed. Then the order is fixed: first the master authenticates the slave and then vice versa. This is also true when a temporary multipoint key is taken into use. It is up to the application whether authentication is performed at other times. These additional authentications do not necessarily have to be mutual. In principle, authentication can be performed arbitrarily many times and in arbitrary order unless the application imposes some restrictions on that.

4.1.4 Encryption Key Exchange

Encryption keys are generated by an algorithm called E_3, which produces a 128-bit result. The inputs to E_3 are:

- current link key

[3] The address of the claimant is included to prevent a reflection attack, where the claimant counter-challenges the verifier with the same challenge, waits for the response, and then answers the original challenge. However, the specifications state that this should not be a problem anyway, since all service request are dealt in the order they were received. Nevertheless, including the claimant address prevents such attacks on not-so-specifications-compliant implementations too. Section 6.2 gives another reason why inclusion of the claimant address is vital for security.

- 128-bit random number
- Ciphering Offset number (COF)

The random number is generated by the master and is supplied to the slave with the control message that requests starting encryption. The last input, COF, takes one of the following values:

- the device address of the master repeated twice, if the link key is a temporary key, or
- ACO produced by the latest authentication, otherwise.

4.2 Encryption

Bluetooth uses an encryption algorithm to conceal the transmitted data from eavesdroppers. The encryption algorithm is a stream cipher called E_0, which is based on Linear Feedback Shift Registers (LFSRs) and a summation combiner [Rue85]. The length of the encryption key can be varied between 8 and 128 bits. The length is negotiated using the Link Manager Protocol (LMP) [Blue03b, Part C].

Encryption is applied only to payload data, its checksum, and a few control fields closely related to that data, called *payload header*. The actual packet header is not protected. There is no real integrity protection, since the checksum is linear Cyclic Redundancy Check (CRC) code and a stream cipher is used.[4] Figure 4.1 depicts how the data are handled. Packet headers are omitted from the figure because they are not relevant to encryption. They are treated separately and even use different kind of error checking and channel coding.

The encryption key produced by E_3 is compressed by a modulo operation in a Galois field of an appropriate size if a shorter key than 128 bits is desired. The stream generator takes a 128-bit key as an input, so the shortened key is expanded back to that size. However, the effective key length is reduced by this operation, of course.

Another important input to encryption is the clock of the master device at the moment when the packet transmission starts. It plays the role of a packet sequence number as it prevents keystream reuse and packet replaying. On the other hand, it also acts as a direction-dependent input preventing keystream reuse for different directions. Transmission of a packet originating from the master always starts at an even-numbered timeslot whereas the slaves start sending at odd-numbered clock periods. Packets occupying several timeslots always occupy an odd number of them. Hence, the clock value at the time the transmission starts uniquely determines the direction, and there is no need for a separate direction bit input as there is in UMTS, for example.

[4] It has been shown that adding public redundancy before encryption, either linear or nonlinear, generally does not ensure integrity, even if the encryption algorithm was secure against chosen ciphertext attack [AB01]. Especially in this case, the attacker knows the characteristic polynomial of the CRC code, and is able to compute the change in the code based on the changes in the plaintext, since an additive stream cipher is used to encrypt them. However, if the polynomial were known only to the sender and receiver, CRC codes could be used to authenticate messages [WC81, Kra94].

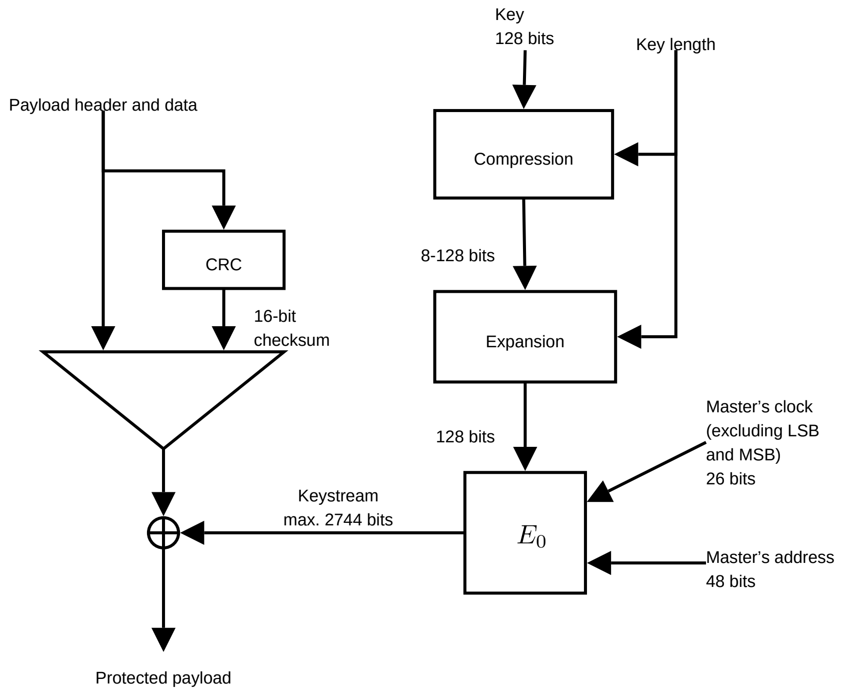

Figure 4.1: Bluetooth packet payload encryption

In Bluetooth, change of the encryption key after clock wrap-around is not mandated. One timeslot takes 625 microseconds of time, so the wrap-around period (2^{26} timeslots) is about 11 hours and 39 minutes.

4.2.1 Encryption Modes

Bluetooth defines three encryption modes:

1. No encryption

2. Point-to-point only encryption

3. Point-to-point and broadcast encryption

Note that it is not allowed to have both mode 2 and mode 3 devices simultaneously in a piconet, since broadcast messages must be either encrypted or unencrypted. In fact, switching between these two modes simply means switching between temporary master keys and combination keys, and deriving appropriate encryption keys. Also unicast traffic is encrypted using the common key in mode 3. Consequently, any device can decrypt all messages sent within the piconet, even if intended only to some other device. In contrast, mode 2 allows for having distinct keys, thus preventing eavesdropping by other piconet members.

4.2.2 Status of Encryption Algorithm E_0

In 1999, Miia Hermelin and Kaisa Nyberg showed how it is possible to recover the initial state of the LFSRs from 2^{64} consecutive keystream bits doing a work of 2^{64} [HN99]. The amount of work has later been reduced to 2^{61} and the required knowledge of keystream bits to 2^{50} [EJ00]. These attacks exploit linear correlations in the summation combiner. Nevertheless, these attacks are of theoretical nature since the LFSRs are reinitialized after each packet and the length of the keystream never exceeds 2744 bits.[5]

At the moment, algebraic attacks seem to be the most effective attacks on E_0. Matthias Krause devised an attack requiring a work of 2^{77} but only 128 consecutive bits of known plaintext [Kra02, Sect. 7]. That amount is eminently realistic for an attacker to obtain but the workload is still prohibitive and equivalent to exhaustive key search of a 78-bit key. Later, Frederick Armknecht and Krause showed how to recover the initial state from 2^{23} keystream bits doing a work of 2^{68} [AK03]. By using a technique called *fast algebraic attack*, which requires some precomputation, the amount of work can be reduced to 2^{55} [Cou03, Arm04b].

The aforementioned attacks concentrate on discovering the initial state of the LFSRs from the keystream bits. Moreover, it has recently been proven that having an effective algorithm for initial state recovery yields an effective algorithm for recovering the secret key [ALP04].

According to Armknecht, recovering E_0 keys using present known plaintext attacks would require about 128 GB of memory and 8 MB of keystream. With present computing machinery, it would take at least 159 years to perform the computations. [Arm04a]

4.2.3 Upgrade of Encryption Algorithm

Even if not breakable in practice, E_0 is of lower security level than AES-based stream ciphers are currently believed to be. Therefore, there are incentives to introduce a stronger encryption mechanism to Bluetooth, preferably based on the AES.

Nevertheless, support for E_0 cannot be removed, in order to provide compatibility with legacy devices. Previous attacks on the GSM system by Barkan *et al.* show that two different encryption algorithms within the same system may interfere in an undesirable manner. Section 4.3 shows how the attack discussed in Section 2.5.2 could be used on the Bluetooth system if amended by another encryption algorithm.

4.3 Key Replay Attack

In the infamous attack on the GSM system by Barkan *et al.*, the attacker can force the victim to reuse a previously used encryption key with the weak algorithm, and then recover the key. Then the attacker can decrypt the previously

[5] The Bluetooth specifications state that the maximum size of payload is 2745 bits. This maximum is achieved by type DM5 packets with 228-byte payload, which maps to 2745 bits due to error-correcting channel coding. However, encryption is applied before channel coding and therefore the maximal-length keystream is used with type DH5 packets having 343-byte payload, which equals to 2744 bits.

generated ciphertext, even if strong encryption has been used. This section shows that such an attack may also be possible in Bluetooth if appropriate counter-measures are not taken. The fundamental cause of the problem is that it is possible to replay encryption keys. Section 4.3.4 presents recommendations for the counter-measures that would be sufficient to allow a second encryption algorithm to be securely taken into use in Bluetooth system.

It should be noted that recovering encryption keys is not the only exploit of the possibility for encryption key replay. For instance, Eric Gauthier presented a key replay attack applicable against the EAP-AKA protocol when a Bluetooth link is used between the victim devices [Gau04].

Let us now assume that a second alternative encryption algorithm is inserted to the Bluetooth system. Then the support for the original E_0 algorithm will be maintained to ensure backward compatibility. Hence, it is necessary to insert a cipher negotiation mechanism to LMP so that the devices can agree on a common algorithm. Moreover, it is natural to impose change of encryption key after change of encryption algorithm to prevent the same encryption key from being used with two different algorithms.

We also make the following additional assumptions about how the new feature is used in Bluetooth system. The assumptions are realistic and in accordance with the current specification.

1. The same E_3 algorithm is used to generate the encryption keys for all encryption algorithms. This is reasonable, since most modern block ciphers, such as AES, use 128-bit keys.

2. The application does not restrict the order of execution of authentication procedures.

3. The link key is not changed often (i.e. it remains the same throughout all sessions involved in the attack).

Finally, we make the general assumption that passive wiretapping and recording of Bluetooth communication as well as active MitM impersonation is possible in Bluetooth. In particular, we assume that the attacker can impersonate the master to a slave, and send control messages to the slave. Note that we do not assume that the master can adjust its clock as is required by Gauthier's attack [3GPP04k, Sect. 2].

It is shown is this section that if these assumptions hold, then it is possible for an active attacker to force a Bluetooth slave device to reuse a previously used encryption key with an encryption algorithm selected by the attacker. In this manner, a situation is created where the attack of Barkan *et al.* works. This violates the requirement that the different encryption algorithms must not pose any threat to each other.

At first, in Section 4.3.1 we consider a simple case involving only combination-type link keys. In Section 4.3.2, we show that under certain conditions this attack can be even easier to perform. Section 4.3.3 discusses whether the attack can be extended to sessions containing point-to-multipoint transmissions.

4.3.1 Basic Attack

In case of point-to-point configurations, which we are now considering, the value of ACO is directly used as COF, the input to the encryption key generation algorithm E_3. If authentication is performed for both parties, the ACO produced by the latest authentication is used. Hence the factors that determine the encryption key are:

- current link key
- master-supplied random number
- challenge supplied by the verifier of the last authentication
- device address of the claimant of the last authentication

The attack works as follows. At first, the attacker records a session that is encrypted by using a strong algorithm. Prior to that, he sees the master supply the last authentication challenge, observes the random number attached to the encryption start request, and saves those messages.

Later, at a moment best suitable for him, the attacker becomes active and impersonates the master to the slave. The old link key is used, so there is no need for mutual authentication. Now the attacker runs the negotiation procedure to take the weak encryption algorithm into use. As explained earlier, a procedure to exchange a new encryption key is performed. It may be possible to use an existing ACO value, as discussed in the next subsection. If a new ACO value is needed, the attacker requests the slave to authenticate itself by sending the previously recorded challenge, as allowed by assumption 2. Being unaware of the real link key, the attacker of course cannot verify the response of the slave, but the result is that the challenge he supplies defines the same ACO, as before. Then the attacker initiates encryption by replaying the random number it recorded from the previous session. The resulting encryption key is identical to the one of the session that the attacker recorded.

It is important to note that if the master is the verifier in the last authentication, the encryption key solely depends on values supplied by him.[6] The slave has then no opportunity to affect the key. This enables the attacker to set up the same encryption key by replaying these values, since by assumption 1 the same E_3 algorithm is used with both encryption algorithms. Now the attacker can try to recover the key by using an attack on the weak algorithm, and then decrypt the ciphertext created using the strong algorithm if he succeeds.

4.3.2 Using Existing ACO

A variation of the attack may be possible if the same ACO is allowed to be used for several encryption key computations. If the same ACO were used, COF would remain constant for long periods of time, just like the link key. Then we are again in the situation where the master is the only one who affects the encryption key. The specifications do not forbid reusing ACOs. In fact, they encourage using the same ACO for several key computations in certain situations. When discussing mutual authentication after a temporary key has been distributed, they say [Blue03b, Part H, Sect. 3.2.8]:

[6] Indeed, the encryption key depends on the current link key the attacker does not know. But because of assumption 3, it is constant throughout the attack and in that sense does not affect the encryption key. As regards to the device address, the same holds.

> The ACO values from the authentications shall not replace the current ACO, as this ACO is needed to (re)compute a ciphering key when the master falls back to the previous (non-temporary) link key.

Therefore, it is highly probable that several implementations do not require a fresh ACO for each encryption key derivation. Attacking on such implementations necessitates only replaying the random number input for E_3, not the authentication challenge, thus rendering assumption 2 unnecessary. It is not even necessary for the attacker to know the last challenge, it is required only that the replay takes place when the ACO value is the same as in the recorded session.

4.3.3 Point-to-Multipoint Configurations

Let us assume that the application allows the master to make the slaves switch to temporary link and encryption keys, and the attacker has recorded a session that contains such encrypted broadcast episodes. It is clear that the attacker is able to recover such parts of the recorded session that were encrypted using a point-to-point key since he can replay separately all authentications and key exchanges he has seen. But could the attacker somehow recover broadcast encryption keys too?

Before broadcast encryption can be started, a new temporary link key is created and transmitted to the slaves, in encrypted form of course. But as mutual authentication always occurs after this, there is no way for the attacker to remain undetected since he does not know the new link key. [Blue03b, Part H, Sect. 3.2.8]

However, there can be applications that constantly use the temporary link key. In that case, the temporary key is never relinquished and the attack works well, just like in the point-to-point case. Note that in this case, the attacker need not know the authentication challenge, but can send any plausible value, since COF is derived from the master's address.

4.3.4 Possible Counter-Measures

Assumption 3 stated that the link key is not changed often. However, if the specifications dictated that the link key must be changed regularly, that would offer some protection against this replay attack. Replaying the challenge and the random number would no longer yield the same encryption key, had the link key been changed. Moreover, as mutual authentication must always occur after change of link key, changing link keys frequently would certainly offer protection against attacks of this kind. Point-to-multipoint applications constantly switching between combination and temporary group keys naturally use this means of protection.

Another possibility to protect against replay attacks is to make the slave always supply the last challenge. LMP definition rules that the slave supplies the last challenge in mutual authentication after the link key has been changed [Blue03b, Part C, Sect. 4.2]. However, this does not by itself prevent the master from initiating new authentication and key exchange procedures immediately after that.

We made the assumption that after each negotiation of encryption algorithm a new encryption key must be exchanged. We assumed that in this process authentication is performed only one way: master authenticates the slave. One might think that requiring mutual authentication would be sufficient to prevent the attacker from replaying an encryption key. However, this is not the case. By impersonating the slave to the real master, the attacker can forward the challenge to the master and get the correct response which it forwards to the slave.

We implicitly assumed that the attacker can freely select the encryption algorithms in protocol negotiation phase. This assumption is based on the fact that currently there is no other integrity protection mechanism than encryption in Bluetooth, and encryption cannot be used before the algorithm has been agreed on. In theory, using message authentication codes based on link keys to protect the negotiation would prevent this attack. However, it would not prevent other types of encryption key replay attacks, such as Gauthier's attack.

Another counter-measure that prevents the same encryption key from being used for two different encryption algorithms is to specify a new different E_3 algorithm for each new encryption algorithm. But again, other types of replay attacks would not be neutralized.

Chapter 5

Security in IEEE 802.15 WPANs

The newest version of the Bluetooth specification is numbered 1.2. Parts of the older Bluetooth specification version 1.1 have been accepted as IEEE Standard 802.15.1 [IEEE02]. After that, the IEEE has approved two other specifications to its 802.15 WPAN standard series. Their security features are discussed in this chapter.

The data transmission rate of Bluetooth radios is 1 Mbps [Blue03a, Part A, Sect. 1.1]. This was considered inadequate for some applications, and therefore the IEEE devised another standard for high-rate WPANs. This standard (IEEE 802.15.3) is not based on the Bluetooth architecture, and its security features are presented in Section 5.1. It allows for transmission rates up to 55 Mbps [IEEE03a, Sect. 5.4.1]. The Bluetooth SIG is also developing a high-rate version of their specification, though the work seems to be progressing slowly.

Interestingly, there is also a specification for low-rate WPANs by the IEEE. This standard, named IEEE 802.15.4, is intended to be used in devices having low complexity and strict power consumption requirements. The maximum data transmission rate is 250 kbps. This system is discussed in Section 5.2. [IEEE03b, Sect. 1.2]

5.1 High-Rate WPANs

The IEEE standard for high-rate WPANs is not based on the Bluetooth architecture, and the security mechanisms are quite different too. This section presents these mechanisms as defined in [IEEE03a].

The standard does not define how security relationships between the devices are formed. It is assumed that there is a mechanism for device authentication and link key establishment provided by upper layers. The link keys are actually called *management keys* in the standard. It is also assumed that there are separate point-to-point and group keys but they can be used simultaneously, that is, without carrying out any switching procedure, in contrast to Bluetooth. Figure 5.1 illustrates what security functionality has been included to the MAC

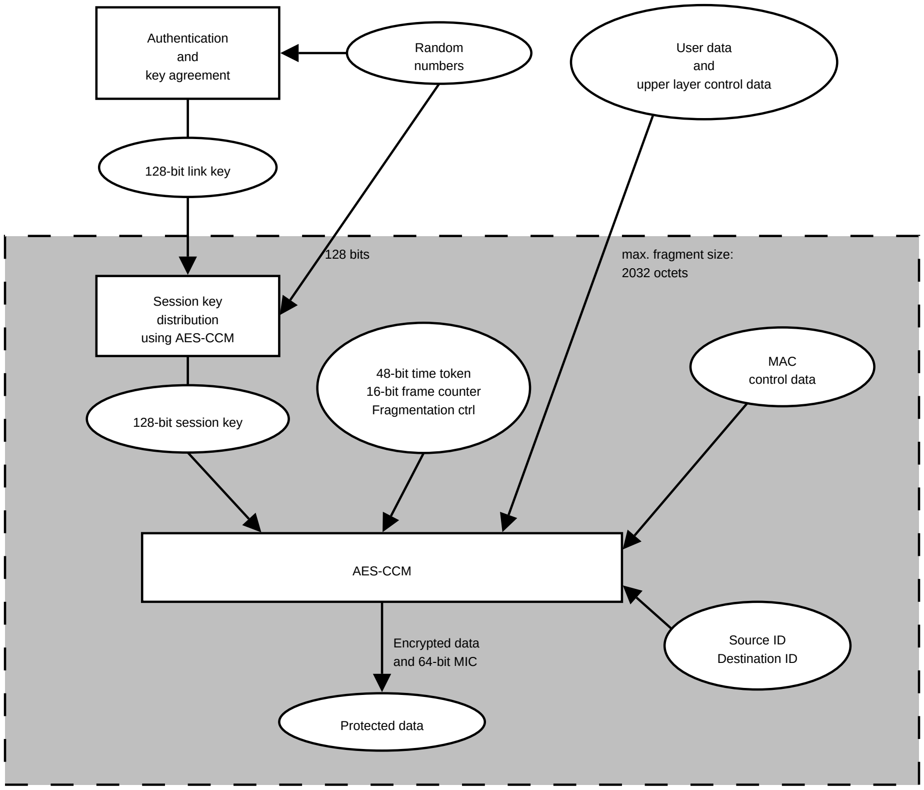

Figure 5.1: IEEE 802.15.3 MAC security functions

layer (denoted by the gray area). The security functions of the Bluetooth link layer (IEEE 802.15.1) are shown in Figure 5.2, for comparison.

Management keys are primarily used to protect distribution of actual data encryption keys, or *data keys.* In addition, management keys are used to protect the integrity of some other control messages. User data are always encrypted and integrity protected by data keys. In this case, protection means using AES in CCM mode, like in WLANs.

Just like there are point-to-point and group management keys, there are two types of encryption keys: those that are used to protect unicast traffic, and group keys intended to protect multicast transmissions. However, the group key can also be used to protect unicast communications if there is no specific security association between the parties. Then this feature provides protection against at least those attackers that cannot get access to the group key. The standard states that encryption keys are generated by a Key Originator (KO), which basically is just one device that is involved in the security relationship in question and capable of performing that task. The generated data keys are encrypted and authenticated using the respective management key (point-to-point or group key) before transmission to other parties.

Encryption is applied only to SDUs and session keys generated by the KO. However, all frames are authenticated, including beacons. The whole content of the frames is authenticated, except for CRC codes that are appended just before

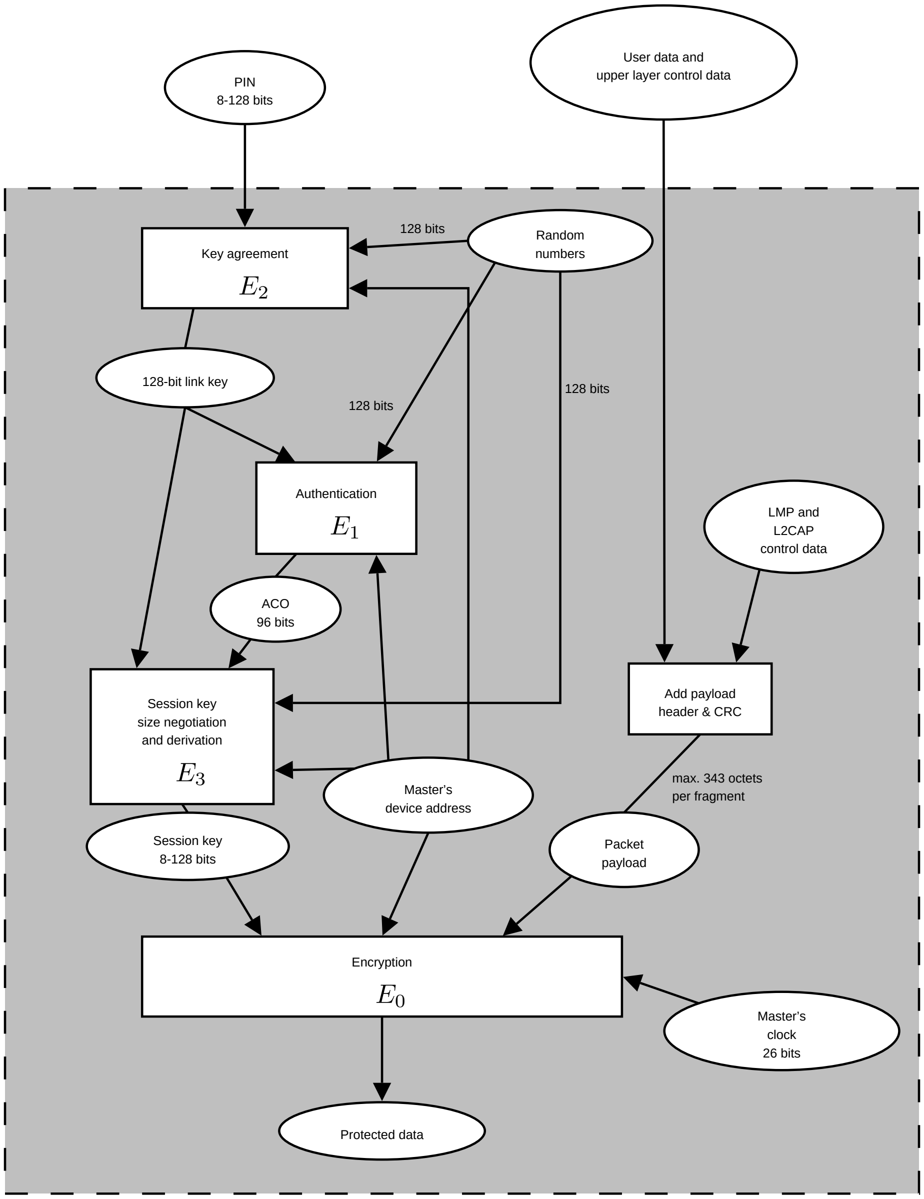

Figure 5.2: IEEE 802.15.1 MAC security functions

transmission for error detection. The nonce handed to CCM depends on the identifiers of the source and destination devices, the 48-bit time token of the superframe[1] during which the protected frame will be sent, and the fragmentation control field, including the fragment number. In addition, a 16-bit counter is included to differentiate the nonce between frames sent to the same destination during the same superframe. Figure 5.3 shows how data frames are handled. As it can be seen, the scheme is pretty similar to the CCMP encryption mode of the new WLAN standard.

Of course, it is not mandatory to use link layer data protection. The specification defines two modes: insecure (Mode 0) and secure (Mode 1). It is up to the PNC to decide which mode is used. If security is used, other devices must establish a security association at least with the PNC before they are granted access to the piconet resources. Moreover, all unprotected frames must be discarded when operating in secure mode. In insecure mode, AES-based protection is not used but the PNC is allowed to maintain an Access Control List (ACL) to restrict access to the piconet.

5.2 Low-Rate WPANs

IEEE 802.15.4 was intended to be used in simple devices. Consequently, the security mechanisms defined by the standard are not complex either. This section presents those mechanisms as defined in [IEEE03b].

Whereas IEEE 802.15.3 has dropped the pairing of the devices outside its scope, IEEE 802.15.4 even omits the session key exchange procedure. The session keys are supposed to be provided by the higher layers, and the standard only defines how to protect the data with a key that has somehow been agreed on. The gray area in Figure 5.4 shows what functionality has been left to the MAC layer.

5.2.1 Encryption and Integrity Protection

IEEE 802.15.4 uses AES as the cryptographic algorithm, and there are essentially three different ways to use it:

- encryption in Counter (CTR) mode [Dwo01, Sect. 6.5]
- integrity protection with Cipher Block Chaining Message Authentication Code (CBC-MAC) [ISO99]
- encryption and integrity protection in CCM mode

If used, the length of the MIC can be chosen to be 32, 64, or 128 bits.

Encryption and integrity protection are applied to both control and SDU frames. Encryption is applied to the payload, if there is any, and MIC is computed over the MAC header and the payload, that is, the whole frame excluding the CRC code. Note that also MAC commands and a certain part of beacon

[1] Superframe means the period between two *beacon frames* sent by the Piconet Controller (PNC, the official term for IEEE 802.15.3 piconet masters). Beacon frames are broadcast to allow devices to synchronize with the piconet. Moreover, each IEEE 802.15.3 beacon frame contains a time token for the beginning superframe that is leveraged in encryption and replay prevention.

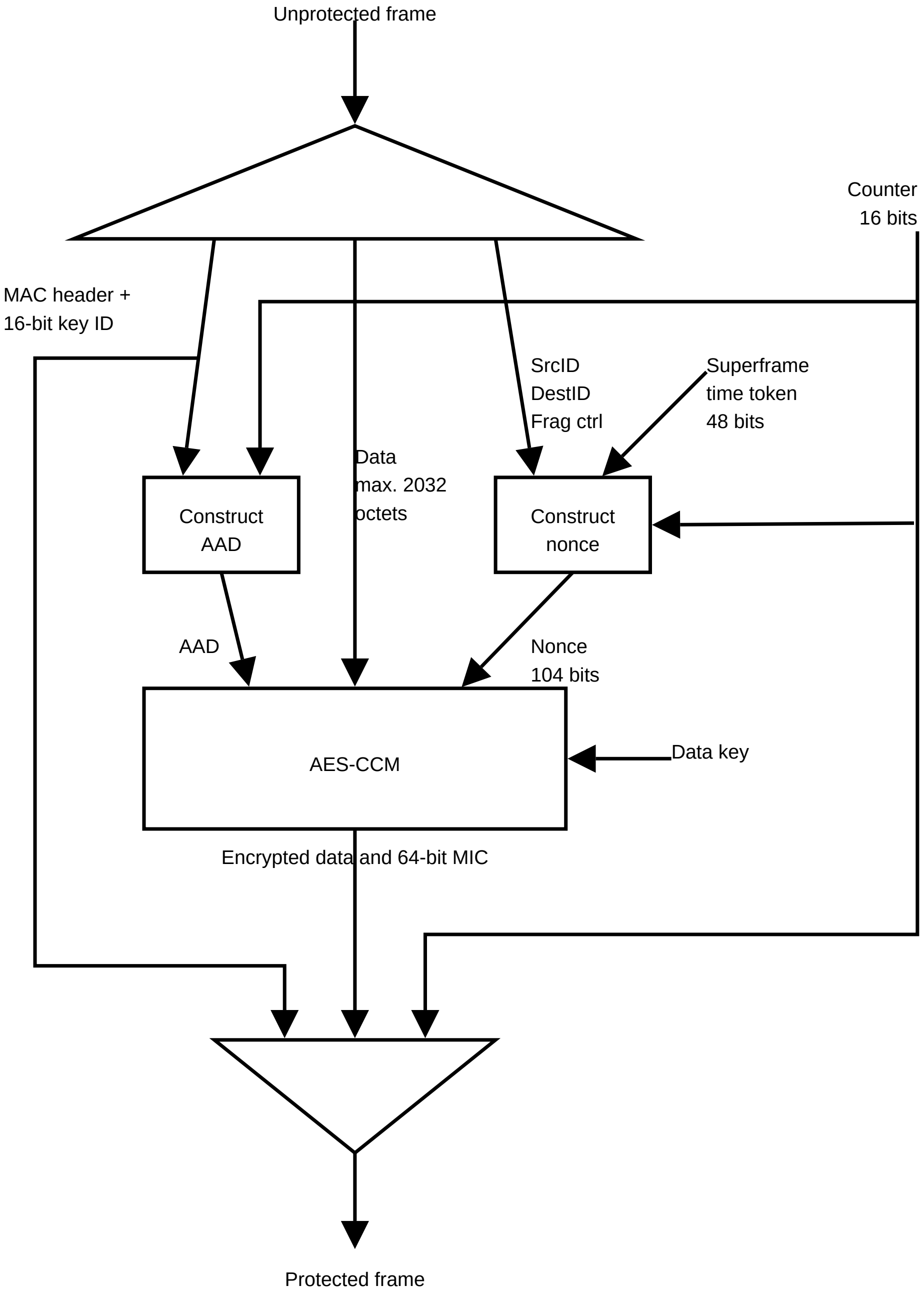

Figure 5.3: Data frame encryption in IEEE 802.15.3 piconets

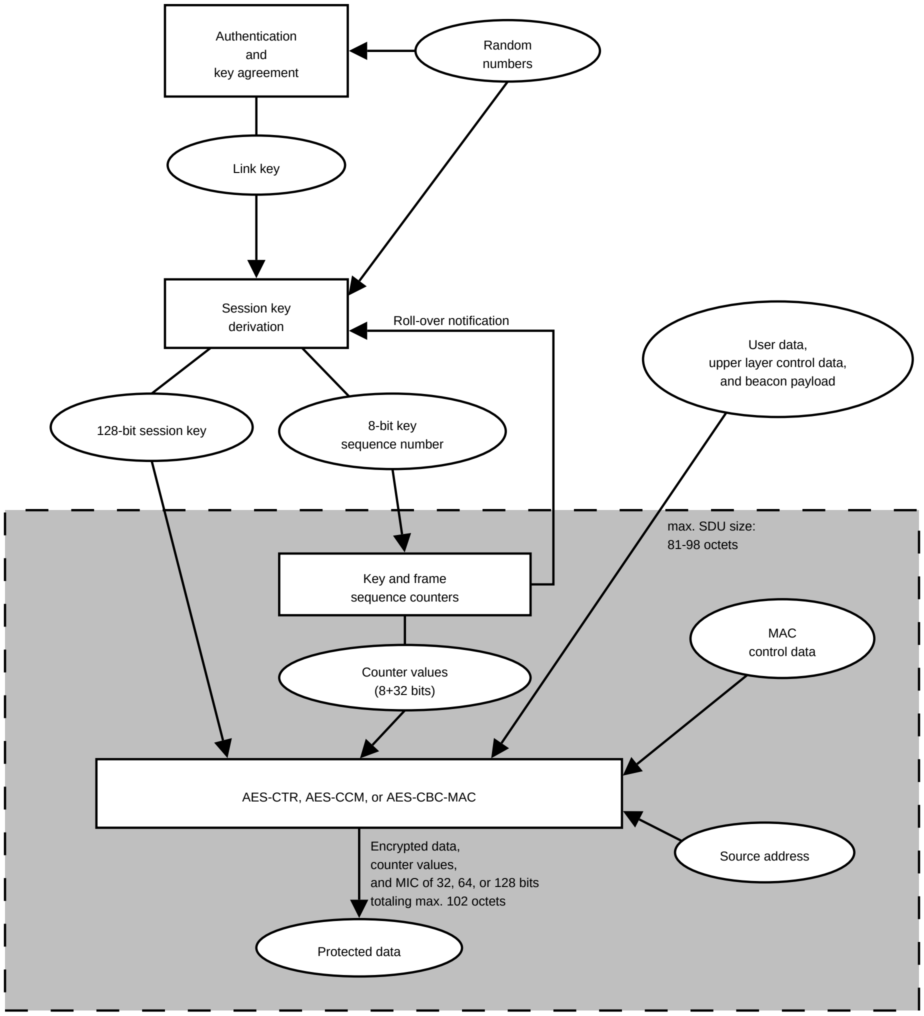

Figure 5.4: IEEE 802.15.4 MAC security functions

frames are considered to be payload.[2] CCM nonces and CTR input blocks are derived from the source device address and two counters. There is a 32-bit key-specific frame counter that is incremented every time a frame is encrypted under that key. In addition, there is an 8-bit key sequence counter, the value of which is given by the upper layers. The sequence numbers are sent alongside with the frame but are not included in MIC computation. In CBC-MAC mode, sequence numbers are not transmitted.

Even though the specification defines no key agreement protocol, roll-over of the frame sequence number is prevented. The MAC layer must indicate an error condition to the higher layer when the counter values have been used up. Apparently, the higher layer is supposed to provide a new session key to the MAC layer in that situation although not explicitly stated in the standard.

Surprisingly, albeit defining different protection modes and MIC lengths, the standard specifies no negotiation protocol that could be used to agree on those parameters, thus leaving also that responsibility to the higher layers. However, it is mandatory to support AES in CCM mode with 64-bit MIC if cryptographic protection is supported at all. On joining the piconet, the devices tell the PAN coordinator whether they implement security.

5.2.2 Security Modes

IEEE 802.15.4 defines three security modes, not to be confused with the AES protection modes mentioned in Section 5.2.1:

1. Unsecured mode
2. ACL mode
3. Secured mode

The first mode is self-explanatory: no security measures are undertaken. The second mode allows using an ACL to prevent unauthorized devices from accessing the services. The access control is solely based on device addresses without cryptographic authentication. The third mode means that AES-based protection is applied. Actually, the first mode can be viewed as a special case of the second mode with an empty ACL. Therefore, the security modes of IEEE 802.15.3 and IEEE 802.15.4 are essentially the same.

5.3 Multi-Radio Systems

Nokia is involved in My personal Adaptive Global NET (MAGNET) research project, funded by the EU Commission. The mission of the project is to enable commercially viable solutions for wireless personal networks that are beneficial in every-day life [MAGN].

One objective of the project is to define a uniform interface isolating the upper layer protocols and the underlying radio access technologies. This interface is called Universal Convergence Layer (UCL), and it is intended to enable

[2] The encrypted part of beacon frames is the *beacon payload*. It is optional and can be used to carry some information provided by upper layers of the Personal Area Network (PAN) coordinator's protocol stack. The upper layers of the other devices may receive this information from their respective MAC layers.

seamless interworking of different radio technologies without user interaction. UCL is to define what information is transmitted between the link layer and the higher layers. Therefore, it would be natural to use it also to communicate security-related parameters, in order to enable utilization of link layer security functions under this multi-radio scenario. [VMS04, Sect. 4.3.1]

Another project called Ambient Networks, also funded by the EU, is also concerned with interworking between different radio systems. Within that project, a Generic Link Layer (GLL) interface is to be specified for adaptation to multiple radio systems. The functions of UCL and GLL seem very similar. [Nie04]

To facilitate incorporating security features to the UCL and GLL interfaces, the IEEE 802.15 family of standards was examined to find out what kind of security information flows through the boundary between the link layer and the higher layers. This section summarizes the results of that examination.

5.3.1 IEEE 802.15.3

The higher layers must provide at least a group management key, and possibly point-to-point management keys if such security associations are desired. Session key request and distribution procedures are initiated by higher layers. In theory, session keys could also be negotiated using some external method instead of the KO approach because commands for session key manipulation exist. [IEEE03a, Sect. 6.3]

All keys are associated an identifier which is transmitted along protected frames. Whenever management or session keys are communicated to the MAC layer, the identifier must be attached. The identifier originates from the same source as the key itself. [IEEE03a]

IEEE 802.15.3 has two security modes. The mode must be specified when a new piconet is started [IEEE03a, Sect. 6.3.3.1].

5.3.2 IEEE 802.15.4

Higher layers must provide the following information:

- security mode
- access rule for the devices not listed in the ACL (accept or reject)
- session key, key sequence number, and security suite (protection mode and MIC length) for each ACL entry
- default values for the aforementioned parameters

The default values are used in broadcast transmissions and with devices not listed in the ACL. [IEEE03b, Sect. 7.4.2]

When trying to transmit a frame but all counter values have been used up, an error message is returned to the higher layer.

5.3.3 IEEE 802.15.1

Higher layers are supposed to provide the link layer with a PIN code to bootstrap security with previously unknown devices. Alternatively, the link key can be agreed on by other means and given to the link layer instead of a PIN. [IEEE02, Chpt. 11]

The encryption mode must also be specified. Switching from mode 2 to mode 3 causes a new temporary link and encryption keys to be generated. The old link key is saved and used again when falling back to mode 2. However, a new encryption key is generated when switching to mode 2 from mode 1 or 3.[3] The master is responsible for setting the encryption mode. As it is possible to change the encryption mode at any time, the higher layers at the slave side are notified when such change occurs. [IEEE02, Chpt. 11]

There are also two on/off-style features concerning connection establishment [IEEE02, Sect. 11.2.7]:

- Require authentication for all connections.
- Use encryption on all connections. If enabled, encryption keys are generated during the connection establishment procedure.

In addition to connection establishment, authentication can happen at any other time too, as requested by higher layers. The higher layers can also request renewal of the link key, which means that a new link key is generated. The related random number exchange is protected by the old link key.

[3] In fact, the IEEE version of the specification does not associate numbers to the encryption modes. However, the same three modes are still defined as in Bluetooth. [IEEE02, Sect. 8.14.3.2]

Chapter 6

Analysis of Security Features

This chapter presents an analysis of some security features of the systems we have studied so far. Session key establishment and replay prevention are discussed in Section 6.1. Capabilities for multicasting and broadcasting, and issues related to them are considered in Section 6.2. Section 6.3 analyzes the input values given to data protection functions, and Section 6.4 discusses an interesting problem related to replay protection and so called *burst transmissions*. Finally, Section 6.5 contemplates why some data are protected whereas other data are not.

6.1 Session Key Establishment

As we have seen, practical communications systems usually have hierarchical key systems. There is always a long-term link key for each security association, which is used to derive more temporary session keys or protect transmission of such keys. These session keys are then fed to the actual data encryption and authentication algorithms.

Table 6.1 shows the relations between the keys in the systems studied in previous chapters, except for IEEE 802.15.4, since it defines only session keys. Properties of WLAN are treated according to RSNA and CCMP throughout this chapter unless otherwise indicated.

Authentication keys are used to authenticate the user or station and somehow derive the long-term link key. Some systems, such as IEEE 802.15.3 and WLAN, do not define the actual authentication mechanisms but rely on higher layer authentication procedures. On the other hand, in GSM and UMTS, the authentication key is itself used as the link key too. In Bluetooth, PINs are used to set up security associations in an *ad hoc* manner. An initialization key is derived from the PIN and that is used to protect the combination key exchange procedure.

A more interesting step is the transition from link keys to session keys. This step has to be designed very carefully. Needless to say, an eavesdropper should not be able to deduce the key if he sees the key exchange messages. In practice, this means that the session key is encrypted using or alternatively

System	Auth key	Session auth key	Session data key	Group auth key	Group data key
GSM	Ki	Ki	Kc	—	—
UMTS	K	K	CK, IK	—	—
WLAN	—	PMK (+ KCK)	TK	KEK, KCK	GTK
Bluetooth	PIN	K_{AB}	$E_3(K_{AB})$	K_{master}	$E_3(K_{master})$
IEEE 802.15.3	—	Mgmt key	Data key	Group mgmt key	Group data key

Table 6.1: Roles of keys

derived from the link key. Moreover, the protocol should be designed so that it is not possible for an active attacker to set up again a previously used session key. As we saw in Sections 2.5.2 and 4.3, GSM nor Bluetooth do not have proper key replay prevention mechanisms, which has bad ramifications. Therefore, careful attention should be paid to key replay prevention when designing new systems.

An attacker trying to replay a session key has two possible objectives. He wants either

- to collect ciphertext encrypted using a specific key and possibly a specific IV for cryptanalysis, or
- to replay previously recorded messages.

Hence, the victim of the attack is the one who encrypts something or verifies authenticity of messages using the key to be replayed.

In the systems we have considered, three replay prevention approaches can be identified:

1. Each party going to encrypt data or verify authenticity of data using a new key is given a means to affect it. The link key should affect the session key derivation in an unpredictable manner.
2. Each party going to encrypt data or verify authenticity of data using a new key is given a means to challenge the contributing parties to prove that they know the link key.
3. Key agreement procedure is protected by sequence numbers. Link key-based message authentication codes are used to verify the messages and their sequence numbers.

The first approach is an implicit key authentication method, whereas the two others are examples of explicit key authentication [MOV96, Sect. 12.2.1].

The 4-way handshake of WLAN uses simultaneously both the first and the second approach. PTK depends on two random numbers, each chosen by one station. On the other hand, the random numbers act as challenges because the responses are protected by MICs using the KCK portion of the newly derived PTK, proving that the PMK is known to both parties. As regards to the GTK part of the exchange, the second replay prevention approach is used, as will be explained in Section 6.2.

4-way handshake is not very economical protocol as it uses two different protection methods, even though either of them would be sufficient. Moreover, the last message of the exchange seems a bit futile. Essentially, the sole content of the message is the MIC, which proves that the authenticator knows the correct KCK. However, has this not already been proven earlier, in the second message? So the last message appears to be a kind of acknowledgement indicating that the third message was received. Another, more efficient approach would be to use only a negative acknowledgement to initiate retransmission when the previous message was lost, if this was the case.

An example of the third approach can be seen in the AKA procedure of UMTS. The challenge that effectively determines the key is protected by AUTN, including a sequence number and an authentication code based on the identification key. The USIM can then verify that the challenge is fresh and was really generated by its own AuC.

6.1.1 Key Replay Attack on IEEE 802.15.3

Also IEEE 802.15.3 uses the third method to protect key distribution. The KO unilaterally determines the key but it must calculate a message authentication code in order to convince the recipients. The superframe number acts as the sequence number. The superframe number is required to be strictly increasing and in the case it is not, the beacon and related frames are rejected [IEEE03a, Sect. 9.3.6].[1]

However, the replay prevention mechanism is not perfect. Consider the following quotation from [IEEE03a, Sect. 9.1.7]:

> To prevent replay of old messages, a strictly-increasing time token is included in the beacon. A DEV may reject as invalid a received beacon with a time token less than or equal to the current time token. In addition, the time token is included in the CCM nonce, as described in 10.2.4, for each secure frame, as described in 7.2, so the integrity check will fail if a frame is replayed in a different superframe. A DEV in a secure piconet maintains two values for freshness. The CurrentTimeToken is the time token value found in the beacon for the current superframe and is used to protect all messages sent and check all messages received during that superframe. The LastValidTimeToken is used by the DEV to ensure that the security of the beacons have [sic] not been compromised.

So what about replaying messages during the same superframe? This possibility will be explored in Section 6.3.2. Indeed, the duration of one superframe is pretty short, so as regards to key replay attacks, this is not a big security problem. But there is another more serious shortcoming in the replay prevention mechanism. Consider the following scenario:

1. Device A joins a piconet, the PNC of which is device B.

[1] Although stated clearly in [IEEE03a, Sect. 9.3.6], there is a clause in [IEEE03a, Sect. 9.1.7] implying that rejecting superframes with an invalid time token is optional because it uses the word *may* instead of *shall*. It is a little dangerous to include this kind of misleading clauses if secure implementations are desired.

2. An adversary, operating device C, observes a key distribution message sent to device A by device B. (Device B is assumed to be the KO.) He also observes the related beacon frame, including its time token T.

3. Device A leaves the piconet.

4. Later, device A wishes to join the piconet again. However, the adversary impersonates device B to device A and replays the beacon frame with time token T. Device B has no other alternative than to initialize LastValidTimeToken to this value. Meanwhile, it may have been a member of another piconet, so it does not remember the correct value for this variable.

5. Device C replays the key distribution message to device A before sending a new beacon.

The consequence is that the MIC is correct and device A accepts the key, assuming that the management key has not been changed. This example shows that using synchronized clocks as sequence numbers does not guarantee freshness unless the key is renewed between losses of synchronization or further actions are taken to handle the situation properly.

6.2 Multicast Communication

All systems we have studied enable sending messages targeted to more than one receiver. In GSM and UMTS, this is limited to base stations broadcasting or multicasting messages to MSs. GSM networks use a special-purpose channel for broadcasting control messages [ETSI00c], but UMTS even allows multicast transmission of user data using the BMC protocol, as noted in Section 2.1.2. Nevertheless, all these messages are unencrypted, hence there are no group key management mechanisms. IEEE 802.15.4 allows encryption of broadcast messages, but since there is no management for unicast keys, there is no such thing for group keys either. It is up to the upper layers to provide such keys.

In contrast, the rest of the systems we have studied do have mechanisms for group key management. IEEE 802.15.3 has completely separate management for point-to-point and group keys. Bluetooth has separate keys for these domains too, and session keys are derived using the respective link key. However, distribution of temporary group keys (master keys) is protected by the current (point-to-point) link key, so they are not completely independent.

A more significant difference between Bluetooth and IEEE 802.15.3 is that Bluetooth specifies that only one link and encryption key can be active and there are procedures for switching between combination keys and group keys. This has some implications on security, and they are discussed in Section 6.2.2.

In Bluetooth, transmission of the temporary group key is protected by the current link key. This would indeed be susceptible to replay attacks unless mutual authentication was not mandated after that, verifying that the master knows the new link key. Now it can be seen that including the claimant address to the authentication response computation is vital. Otherwise, as several slaves share a common link key, a master impersonator could perform a reflection attack by forwarding challenges from one slave to another and receiving the correct response without knowing the link key. However, executing the mutual authentication procedure is not good enough, since the transition from the temporary group key to a group encryption key is not protected.

WLAN has a bit similar hierarchy with respect to group keys. Transmission of GTKs, the group encryption keys, is encrypted by the KEK portion of the PTK. The GTK distribution part of the 4-way handshake obeys the second replay prevention method. The message containing the encrypted GTK is protected by a MIC calculated under the KCK. The random number (SNonce) supplied by the supplicant in the second message acts as a challenge, since it affects the KCK.

In addition, GTK can be renewed without changing the PTK, that is, re-executing the 4-way handshake. The group key handshake protocol uses the old KCK and KEK to protect transmission of the new GTK. Since no new challenge is supplied by the receiver, it might seem that replay attack would be possible. However, group key handshakes are protected by 64-bit sequence numbers [IEEE04, Sect. 8.5]. In other words, they use the third replay prevention method of Section 6.1.

It is interesting that there is a separate handshake procedure for changing the GTK but changing the TK implies changing also the GTK. This feature would be justified if multicast packets were sent significantly more frequently than unicast packets. However, this is probably not the case in most applications, and this procedure just complicates the RSNA implementation.

6.2.1 Common Group Keys

An important thing to discern about all the WPAN systems discussed here is that they use common group keys. This enables all group members to impersonate any other member of the group when sending multicast messages. This is even possible in WLAN although each station has its own GTK. But since GTKs are communicated to other stations to allow them to decrypt and verify authenticity of messages, they can also use it in impersonation attacks.

Solving this problem would mean enabling others to decrypt and verify authenticity without the ability to encrypt or calculate authentication codes. In practice, this would mean employing public key cryptography [Sti02, Sect. 5.1], which is not usual in radio systems. Of course, public key methods would call for far more computing capacity from the hardware than the symmetric methods.

Since this is the case, it may seem odd that there are separate group keys for each WLAN station. Maybe the designers thought that it would help migrating to public key algorithms in future. Or maybe they just wanted to avoid a complex protocol for common key agreement. Bluetooth and IEEE 802.15.3 networks are centrally coordinated piconets and the group keys are issued by the coordinator, but in *ad hoc* WLANs, there is no such coordination, so common key agreement is not a trivial problem. In infrastructured mode, there is a common AP, but in that case, only one group key is actually used as pointed out in Section 3.2.

6.2.2 Bluetooth Master Keys

If temporary master keys are used to protect Bluetooth broadcast transmissions, also unicast messages must be encrypted using the common key. This is justified by assessing that some devices might be incapable of switching between the keys in real time, after examining the packet header [Blue03b, Part H, Sect. 3.2.6]. In my opinion, this is quite a big sacrifice in terms of security, since it allows eavesdropping of unicast traffic by other members of the piconet for the time the temporary key is enabled, that is, as long as encryption mode 3 is used.

Can this problem be solved, assuming that there really are such devices with limited capabilities? Consider how such devices would be used in the IEEE 802.15.3 system. In that system, it is allowed to use the group encryption key if no point-to-point security association exists. Using these keys for all unicast traffic is equivalent to the Bluetooth encryption mode 3. Limited devices can do so, but they are still subject to eavesdropping by other piconet members. However, the other devices can still use point-to-point keys if they choose to do so, whereas in Bluetooth, all devices are deprived the ability of using such keys due to the *possibility* of having some limited-capability devices in the piconet.

To defeat this disadvantage, I propose that fourth encryption mode be defined. It would be named "Point-to-point and broadcast encryption using distinct keys," and it would allow using different encryption keys for unicast and broadcast messages. The point-to-point key could be derived from the old link key that must be saved anyway, for example. This would be fully compatible with the current specifications. However, a negotiation mechanism should be inserted to the LMP, by which the master could find out whether the device is capable of using two encryption keys simultaneously.

6.3 Inputs to Protection Functions

As we have seen, the outputs of encryption and integrity protection functions depend on various inputs, in addition to the session key and the data to be protected. The primary reason for this is to avoid using the same keystream more than once. As noted in Footnote 6 of Chapter 2, reusing keystreams reveals partial information about the plaintext to an eavesdropper. Another reason is message replay prevention. This section analyzes what kind of inputs should be used to achieve security.

6.3.1 Counters versus Random IVs

There are systems employing IVs that are not required to be strictly increasing. One example is WEP, the original WLAN data protection protocol. Some implementations used a counter value as the IV, but that was not mandatory. Using mere random numbers would have been perfectly compliant with the specifications.[2]

However, using counters has a couple of advantages over using random IVs. One of them is that the same counter can also be used to prevent active attackers from replaying previously recorded frames, in addition to ensuring keystream unicity. If the integrity checking code depends on a strictly increasing counter

[2] In fact, it was not even required that the IV should be changed for each frame. It was only stated that IV be changed adequately often, depending on the protocols used at upper layers. This fact alone is in direct contradiction with the clause in the specifications stating that WEP is "reasonably strong." [IEEE99, Sect. 8.2]

included in frames, replayed frames can be detected and discarded. Even if integrity protection is not applied, counter-dependent encryption makes it difficult to replay messages.

Another reason why counters are better is that for the same level of security, you would need a random IV that is much longer than the counter would be. This is due to the birthday paradox. If we use counter having length of N_C bits, we know that the same value will occur exactly after 2^{N_C} messages. Thus, $2^{N_C}+1$ consecutive values are needed for a collision to occur. But when using random IVs, we can never know when a specific value appears next time. It can only be shown that if we have q randomly chosen values from M different possible values (with uniform distribution), then with probability ϵ there are at least two values that are equal to each other when

$$q \approx \sqrt{2M \ln \frac{1}{1-\epsilon}} \tag{6.1}$$

The proof can be found in [Sti02, Sect. 4.2.2]. So if we want to achieve the same level of security with random IVs as with an N_C-bit counter, let us substitute q by 2^{N_C}. Let N_{IV} denote the length of the corresponding IV, that is $\log_2 M$. After these substitutions the above equation takes the following form:

$$2^{N_C} \approx \sqrt{2^{N_{IV}+1} \cdot \ln \frac{1}{1-\epsilon}} \tag{6.2}$$

Taking base 2 logarithms, this yields

$$N_C \approx \frac{1}{2}\left(N_{IV} + 1 + \log_2 \ln \frac{1}{1-\epsilon}\right) \tag{6.3}$$

Solving for N_{IV} gives

$$N_{IV} \approx 2N_C - 1 - \log_2 \ln \frac{1}{1-\epsilon} \tag{6.4}$$

Had we used counters, we would still have probability zero for collisions when $q = 2^{N_C}$. It is not possible to attain this level of security by any finite-length random IV because

$$\lim_{\epsilon \to 0+} N_{IV} = 2N_C - 1 - \lim_{\epsilon \to 0+} \log_2 \ln \frac{1}{1-\epsilon} = \infty$$

Anyway, even ignoring the last term in Equation (6.4), it can be seen that the length of a random IV must be at least about two times that of a counter. Hence, counters are more economical than random IVs and should be preferred in future systems. However, it should be noted that certain modes of operation require IVs to be randomly chosen as explained in Chapter 7.

All systems we have studied use counter-based initialization. Table 6.2 summarizes what additional inputs to protection functions are used in the systems studied in previous chapters. It also shows the role each input is playing. The remaining part of this section concentrates on analyzing the purpose of these additional inputs.

System	Channel	Counter	Direction	Counter skew prevention	Other
GSM	Physical layer TDMA frame #		Split keystream	—	—
UMTS	Bearer ID	Packet #	Input to *f8* & *f9*	FRESH (*f9* only)	—
WLAN	Priority	Packet #	Trnsmtr addr	—	—
Bluetooth	—	Master's clock		—	Master's addr
IEEE 802.15.3	—	Superfrm #, Packet #, Frag ctrl	Src ID, Dest ID	—	—
IEEE 802.15.4	—	Key #, Packet #	Src addr	—	—

Table 6.2: Additional inputs to data protection functions

6.3.2 Packet Counters

All these systems include some kind of counter in keystream generation to achieve the aforementioned security goals. It is either a packet sequence number transmitted in cleartext with the packet, or a physical layer frame number or another value known to both communicating parties by synchronization of clocks.

IEEE 802.15.3 uses counters of both types. The superframe time token is determined by synchronization whereas the packet number, officially called Secure Frame Counter (SFC), is prepended to the encrypted payload [IEEE03a, Sect. 7.2]. Although not mandated, SFC can be reset to zero when the superframe changes [IEEE03a, Sect. 7.2.7.3]. The specifications are obscure about whether SFC values should be distinct for different fragments of a single SDU. When discussing the nonce value, they say [IEEE03a, Sect. 10.2.4]:

> In order to preserve the security of the symmetric algorithms, this nonce shall be unique. As a result, the DEV shall not reuse any 2-octet sequence number within a single superframe that is intented for a particular DEVID (as this would cause a repeated nonce).

However, the nonce depends also on the fragmentation control field in the MAC header. This field includes a third counter, namely the fragment number. Hence, it seems that the same SFC value could be used. Including the fragment number to the nonce implies that using the same SFC does not yield identical nonces for different fragments of one SDU.

As speculated in Section 6.1.1, it might be possible to replay IEEE 802.15.3 frames during the same superframe under which they were sent. The specifications state nowhere that the value of SFC also be checked, to ensure that no value has been used more than once. However, the fragmentation control field affecting the nonce includes also an SDU sequence number that is used to detect duplicate transmissions of SDUs or fragments thereof [IEEE03a]. Inclusion of such data to the nonce might seem to exclude any possibility for packet replay.

However, the length of that sequence number is only 9 bits, and it might well roll over during a superframe. If that was not possible, it would have been pointless to introduce the SFC at all. In fact, in [IEEE03a, Sect. 10.2.4] it is said that keystream unicity is guaranteed unless more than 2^{16} frames are sent to a single destination during a superframe, so it is reasonable to assume that the number of frames is not limited to 2^9. This being the case, it is strange that there is no protection against SFC replay. Such functionality could easily have been implemented in several ways, as will be shown in Section 6.4.

In addition to having strictly increasing packet number counters, the WLAN RSNA definition and IEEE 802.15.4 require that no counter value is used twice. In practice, this means changing the session key when the counter rolls over. In theoretical point of view, this is the right thing to do, to absolutely refrain from reusing counter values with the same key. In a way, the packet counter measures the usage of the corresponding key. The other systems do not have this kind of property, however. In IEEE 802.15.3, the functionality of measuring key usage would be attainable if the counter was longer than 16 bits and it was not allowed to reset it when the superframe changes [IEEE03a, Sect. 7.2.7.3]. If that was the case, the time token could be removed from the nonce, had the replay prevention mechanism been designed properly.

The IEEE 802.15.4 specification does not too much elaborate the very essence of the key sequence counter. Having read the definition of that parameter in [IEEE03b, Sect. 7.6.1.8], one can think that it could be used as an extension of the frame sequence number in case of roll-over. But why then have two distinct counters instead of one 40-bit counter? Or perhaps the key sequence counter could be used to prevent key replay attacks on a higher layer key distribution protocol, together with an authentication code. However, such functionality should be included to the respective layer to promote layer independence.

In WLAN headers, there is a field named Key ID, which is used to facilitate the transition to a new group key or allow switching between several WEP keys [IEEE04, Chpt. 8]. Footnote 8 of Chapter 2 mentioned KSI, which is there to help avoiding key inconsistencies after UMTS handovers. In GSM, there is a value with a similar function, namely the Ciphering Key Sequence Number (CKSN) [ETSI00a]. However, the IEEE 802.15.4 key sequence number does not compare well with CKSN, KSI, or WLAN Key ID, since it is completely omitted in CBC-MAC mode. Hence, its purpose may have something to do with keystream reuse avoidance.

The fact that the sequence numbers are omitted in CBC-MAC mode of IEEE 802.15.4 implies that there is no frame replay protection in that mode. The attacker cannot forge an arbitrary message but he can replay any message he has previously seen.

6.3.3 Maintaining Parallel Counters

Another component that seems to be present in every system is some direction-dependent action or value supplied to the keystream generator, or the integrity function if there is one. Various methods are used. GSM uses different portions of the keystream for uplink and downlink, whereas in UMTS, there is a special input bit to the cryptographic algorithms. In Bluetooth, no separate action is needed, since the clock used as a counter input determines the direction of transmission. In addition, only one device can send during a single timeslot [Blue03b, Part B, Sect. 8.6], so common temporary keys are neither a problem.

WLAN uses the transmitter address field to differentiate the nonce between the directions. This is essential only in unicast transmissions, since for group transmission every station has its own GTK. Similarly, IEEE 802.15.3 and IEEE 802.15.4 use the source identifier to achieve the same effect. However, IEEE 802.15.3 also takes the destination identifier into account. The reason for this is that in IEEE 802.15.3, it is allowed to use the common group encryption key if a point-to-point key does not exist. Therefore it may happen that the same encryption key is used with packets sent to several different destinations.

In UMTS, there is an input called Bearer ID. It is necessary because the frame counters are maintained separately for different radio bearers, and thus the same values of counters can be used more than once on different bearers. In GSM, the multitude of channels is not a problem because a single physical layer frame cannot contain data from more than one channel [ETSI00c]. As pointed out in Section 3.3, WLAN frame priority classes form logical channels having separate replay detection counters, and that is why the priority is used in nonce construction.[3]

Now comparing this to the direction-dependent inputs, we can see that the fundamental reason for having such inputs is that the frame counters are maintained separately for both directions. Hence, they serve exactly the same purpose as the Bearer ID input in UMTS. All these inputs just distinguish between different counters used under a single encryption key, or hierarchies of such counters.

Why then have several parallel counters at all? There are two possible approaches to multiple channel access in radio systems: contention-based and centrally coordinated scheme. There are systems that use only one of these mechanism. For example in Bluetooth, only the centrally coordinated method is present. On the other hand, some systems, such as WLAN [IEEE99, Chpt. 9], IEEE 802.15.3 [IEEE03a, Sect. 8.4], and IEEE 802.15.4 [IEEE03b, Sect. 7.5.1], use both. When contention-based access is allowed, there cannot be shared frame counters between different stations.

Another reason might be layer independence. In UMTS, there clearly is a common reference clock at the physical layer, but as protection algorithms can be applied two or three layers above the physical layer, it would not be convenient to use such information. Moreover, having separate counters for different radio bearers makes parallel protocol instances independent of each other.

Although necessary to have separate counters at each WLAN station, it would not be necessary to have separate counters for each security association as required in [IEEE04, Sect. 8.3]. In principle, a transmitter could use a common counter for all destinations, possibly made a few bits longer. However, the drawback would be that all session keys would have to be renewed when the counter rolled over, even if some of the keys were not used frequently, causing superfluous key exchange traffic. Thus the third good reason for having parallel counters is that distinct counters for different session keys can be used to determine when the key be changed.

[3] If priority was not used in nonce construction, an attacker might try to replay a frame in a lower priority class. However, this could be prevented also by authenticating the priority class information itself. But as the specification does not define how that information is signaled, the approach used here is probably the better alternative.

Fourth reason for parallel counters was already discussed in Section 3.3. Having several priority classes necessitates reordering of frames, which in turn necessitates separate counters for each class. It is worth noting that the WLAN specification rules that there is only one transmission counter that is common for all priority classes [IEEE04, Sect. 8.3]. In theory, there could be separate transmission counters too. That would make it possible to use the session keys even longer if the classes were used evenly. Indeed, having more counters would complicate the implementation, which was probably the reason for the decision. But I see no reason why that could not have been made optional.

IEEE 802.15.3 does not mandate maintaining a distinct counter for all destinations although it is possible even with the group key due to the destination identifier input to encryption [IEEE03a, Sect. 10.2.4]. There seems to be no reason for using a distinct counter for each security association because a 16-bit counter is not long enough to be used as a basis for determining when to renew the key. Replacing the destination identifier by additional counter bits would have been more efficient. Then there would be more nonces available for a single destination if the other destinations were not sent to so often. In fact, IEEE 802.15.4 allows using a common default encryption key[4] and solves the problem in this manner, that is, by using a common 32-bit counter [IEEE03b].

It might be, however, that the source and destination identifiers are included also to impede precomputation attacks on encryption using the group key. Now an attacker has to compute separate lookup tables for each source–destination pair he wishes to attack on, instead of just one common table. More information about attacks of this kind can be found in [WHF03, Sect. 5].

6.3.4 Counter Initialization and Reset

Usually packet sequence number counters are initialized to zero when a new session key is established. However, UMTS allows UEs to unilaterally choose the initial counter values after handovers, arbitrarily many times during the lifetime of the key. This would be dangerous if the FRESH parameter, a random value chosen by the RNC, was not there. Otherwise an active attacker could rewind the counter and replay previously recorded messages, but now the 32-bit value of FRESH is different from the previous value with an overwhelming probability and replay attacks are not likely to succeed.

So if the message originator is able to decrement the counter values, the *receiving* party must be given an opportunity to affect the value of the integrity protection function, at least as often as counter values are adjusted. As regards to decrementing the encryption counter, the situation is just the opposite. The message *sender* should provide randomness to keystream generation if the receiver could decrement the counter. This is necessary to prevent the receiver from collecting two or more ciphertexts encrypted using the same keystream. Note that this is not needed in UMTS, since performing a successful handover by itself apparently requires being able to calculate valid integrity codes under the new FRESH value [3GPP04h].

Also GSM allows continuing the use of the same session key after a handover, since the security context can be communicated between MSCs via a common

[4] This default encryption key is the same key that is used to protect broadcast frames, including beacon frames.

VLR, and also within an MSC area [ETSI00e]. GSM uses the clock of the base station as the frame counter, the MS's clock being synchronized with it. So what can be said about handovers? Is it possible for the new base station to skew the clock? The answer is yes, since the clocks of different base stations are not required to be synchronized with each other. In fact, the clocks may differ even in different *cells* served by a single base station [MP92, Sect. 9.4.2.3]. By considering the rule stated in the preceding paragraph, a problem can be identified. An attacker impersonating a base station can rewind the clock and receive ciphertext encrypted by a previously used keystream.

The other systems we have studied do not define handover mechanisms — at least yet. But if such mechanisms are introduced some day, which seems pretty likely especially with WLAN, this is an issue not to be forgotten.

Note that the non-existence of a FRESH-style input parameter is the fundamental reason for the flaw in the IEEE 802.15.3 key distribution protocol that was revealed in Section 6.1.1. There is no actual handover procedure, but the loss of synchronization due to leaving the piconet for a while causes a similar effect. If devices were allowed to submit a random parameter affecting the MIC calculation every time they joined the piconet, a KO impersonator would not be able to replay the key. It is important to discern that the attack is not limited to replaying key distribution messages. If the attacker manages to rewind the time token and set up the same session key, nothing prevents him from replaying SDU and command frames too. Moreover, by these actions the attacker can force the other party to encrypt messages using previously used keystream.

The attack discussed above is also well applicable on Bluetooth. As shown in Section 4.3, replaying the encryption key is possible. Indeed, the specification forbids the master to adjust the clock during the existence of the piconet [Blue03b, Part B, Sect. 1.1]. The slave can possibly detect drastic changes in the clock of the master, but after visiting another piconet, there is no way for the slave to know if the clock has been skewed. Requiring mutual authentication does not help. Since the authentication function E_1 does not depend on the clock, the attacker can perform the reflection attack, that is, impersonate the victim to the real master to obtain the correct response. After that, the attacker is free to replay any previously recorded frame at the appropriate timeslot. He may also receive packets encrypted using an old keystream. Note that Gauthier's attack mentioned in Section 4.3 is a special case of this attack.

We can see that the objectives of these counter skew attacks are to

- obtain partial information (xor sums) of plaintexts, or
- replay previously recorded messages.

In fact, these are quite similar to the goals of key replay attacks listed in Section 6.1. The only difference is that an attacker performing key replay attacks might have also other means for cryptanalysis than exploiting partial knowledge of plaintexts.

6.3.5 Classification of Inputs

There is one seemingly useless input to the Bluetooth encryption algorithm. Using the master's device address in Bluetooth encryption does not affect anything, since it is a constant and thus is a useless input. With this exception, we can see that additional inputs are all about counters. Counters prevent keystream reuse and message replay when properly amended by some additional variables. We can see that each useful input value is of one of the following types:

Counter (CTR) that is incremented after one frame or fragment thereof has been sent, or when another counter rolls over. Alternatively, counters may be based on synchronized clocks.

Counter Identifier (CID) that combined with the other inputs of this type uniquely identifies the set of counters, the values of which have been used as input.

Counter Skew Guard (CSG) that provides protection against replay attacks when one communicating party is able to unilaterally reinitialize counters during the lifetime of the session key.

Table 6.3 shows the roles of the different useful inputs according to this classification.[5] The direction "input" to the GSM keystream splitter is classified as a counter. As an uplink timeslot always follows the downlink timeslot with the same number after a period corresponding to three timeslots [ETSI00c], the direction can be viewed as a one-bit subcounter of the actual frame number. As regards to IEEE 802.15.3, it is good to note that the CIDs distinguish only the packet and fragment counters. The superframe counter is common to the entire piconet.

Not all inputs that do not belong to the aforementioned classes are completely useless. For example in CCM mode, the length of the nonce cannot be chosen independently but rather depends on the choice of another parameter [WHF03, Sect. 2]. In this case, it is better to include some variable data instead of a constant to impede precomputation attacks mentioned in Section 6.3.3.

6.4 Selective Acknowledgement and Retransmission

We have seen that using message authentication codes does not by itself ensure that the message was sent by the correct party. At most, a valid code indicates that a similar message has been *generated* by that party at some point of time, but it does not exclude the possibility of an attacker having replayed the message.

It is interesting that the need for protecting against replay attacks is mentioned in [Dwo04, Sect. B.1]. Nevertheless, it is not said that sequence numbers could be used to defend against message replay. It is neither said that CCM nonces could be constructed using sequence numbers, not to mention that the same numbers could be used for both purposes, although this is the most commonly used approach in the recent specifications.

However, requiring strictly increasing sequence numbers for replay prevention is somewhat conflicting with the idea of burst transmissions, meaning that

[5] In addition to the master's address input to the Bluetooth encryption, the key sequence number of IEEE 802.15.4 has also been omitted, since no security reasons for its inclusion were found.

System	Input	Type
GSM	Frame number	CTR
	Direction	CTR
UMTS	Packet number	CTR
	Direction	CID
	Bearer ID	CID
	FRESH	CSG
WLAN	Packet number	CTR
	Transmitter address	CID
	Priority	CID
Bluetooth	Master's clock	CTR
IEEE 802.15.3	Superframe number	CTR
	Packet number	CTR
	Fragmentation control	CTR
	Source identifier	CID
	Destination identifier	CID
IEEE 802.15.4	Packet number	CTR
	Source address	CID

Table 6.3: Classification of additional inputs

several packets are sent successively, without requiring acknowledgement between them. After the burst, the receiver indicates which packets were not received successfully, resulting in retransmission of those packets. This kind of selective acknowledgement and retransmission puts the packet numbers out of order. One possible solution is to renumber the packets before retransmission, but that would necessitate re-encryption of the packets too if encryption depended on the same packet number. If separate numbers were used, the integrity code would still have to be recomputed. Due to the additional workload and power consumption caused by re-encryption or MIC recomputation, this solution is often undesirable.

We have studied four systems having a real message authentication mechanism. Two of these, WLAN [IEEE99, Chpt. 9] and IEEE 802.15.4 [IEEE03b, Sect. 7.5.6], do not involve burst-type transmissions. The RLC protocol of UMTS provides selective acknowledgements, but as replay protection is performed at the RRC layer, different sequence numbers are used for these purposes, thus eliminating the need for re-encryption [3GPP03b, 3GPP04h]. This works because the original packet order is restored before replay filtering.

As mentioned in Section 6.3.2, the intra-superframe replay prevention mechanism of IEEE 802.15.3 is deficient. It would have been possible to make it more complete for the following reasons, even either of them being sufficient:

- the SFC is incremented on frame retransmission, and
- the order of SDUs is preserved by the MAC layer (although they are necessarily not transmitted in that order).

Incrementing the SFC implies that the standard replay counter solution would work, one counter per sender being adequate. [IEEE03a]

However, incrementing the SFC necessitates re-encryption of retransmitted frames. Had this requirement been omitted, incorporating replay protection would still have been relatively easy due to the SDU order preservation property. More precisely, there can be different *streams* of data, such that the order is preserved within them, whereas the order is not necessarily preserved with respect to other streams [IEEE03a]. The streams form logical channels in a similar fashion to the WLAN priority classes.[6] Therefore, defining one SFC replay counter for each stream, just like they have their own SDU number counters, would have prevented SFC replay. If this solution had been used, the replay filtering should have been performed after order restoration, of course. Note that the stream index (identifier) field is part of the MAC header [IEEE03a, Sect. 7.2] and thus authenticated, so it would not have been possible to replay frames on different streams. Nevertheless, the most elegant solution would perhaps have been to make the SDU numbers longer than 9 bits, forget separate SFCs, and include the stream index to the nonce as a CID-type input, to prevent message replay on different streams.

Note that if incremented on retransmissions, SFC would no longer verify the correct position of the SDU in the stream. Therefore, inclusion of the SDU number to the nonce or at least authenticating it would be important in order to enforce SDU order preservation.

6.4.1 Replay Detection Algorithm with Memory

The aforementioned solutions do not work, if in-order packet transmission is not available. What is the correct way to handle replay protection of burst transmissions in such a case? Consider the following solution: In addition to maintaining a register containing the SFC of the packet last received (denoted by N), a b-bit register s is used to keep track of the used SFC values. Note that implementing the selective acknowledgement protocol itself requires using a similar register to keep track of the lost packets, so this is not a big tradeoff. We denote the ith bit of s by s_i, and $s \ll n$ refers to the *shift left* operation, after which s_i contains the value stored at s_{i-n} before the operation if $i \geq n$. Otherwise s_i is set to 0. Suppose that N and all s_i are initialized to 0. On receiving a packet with SFC p, the algorithm shown in Figure 6.1 is executed, in order to decide whether the packet should be accepted or rejected. The algorithm returns ACCEPT when the SFC is greater than any previous SFC value, or SFC is within the window of size b and it has not been used before. Otherwise, the algorithm returns REJECT. We will call this solution the Replay Detection Shift Register (RDSR).

Note that it is important to verify the authenticity of the received frame before the registers are updated. Otherwise there would obviously be room for a DoS attack. The authenticity can be verified before executing the RDSR algorithm but it might be more efficient to do it just before updating s and N. If the authenticity check failed, the algorithm execution should be aborted and the frame rejected. This problem that the SFC value needs to be verified efficiently

[6] The specifications use term *stream* when referring to isochronous data streams. In addition, the normal asynchronous data flow forms an additional logical channel although not called a stream. However, in this discussion, term *stream* refers also to this channel.

```
if $p > N - b$ then
    if $p > N$ then
        $s \ll (p - N)$
        $N \leftarrow p$
    endif
    if $s_{N-p} = 0$ then
        $s_{N-p} \leftarrow 1$
        return Accept
    endif
endif
return Reject
```

Figure 6.1: RDSR algorithm

before replay counters are updated is not unique to the RDSR algorithm but common to all replay detection mechanisms based on sequence numbers.

6.5 Scope of Encryption and Authentication

Having examined some systems, we have seen that some data are encrypted and authenticated, whereas other data might not be protected at all. Yet there are data that are authenticated but not encrypted and vice versa. Table 6.4 shows how different types of data are treated in the systems we have considered. Note that data related to session key and group key exchange are omitted in the following discussion, since they were already analyzed in Sections 6.1 and 6.2. In addition, messages and data related to error detection and correction as well as synchronization are not considered here either.

The common thing to all systems is that the data provided by upper layers (SDUs) can be encrypted. Usually, authentication is also applied if an appropriate mechanism exists in the system. However, UMTS does not apply authentication to user data although there is a mandatory-to-implement protection function. Only control data transmitted using RRC are authenticated.

Link layer commands are usually authenticated, or at least encrypted if authentication is not used in the system. WLAN is an exception, since authentication and encryption is applied only to data frames, even if RSNAs are used. This may be due to the attempt to make TKIP resemble WEP as much as possible, in order to facilitate upgrading legacy devices to use TKIP. Although there is no protection for MAC commands, one could argue that security has not been cut down too much because by forging such commands, an attacker could only perform DoS attacks. Such attacks could also be executed using radio jamming anyway, as noted in Section 1.2.

On the other hand, in mobile telephony systems, it is more important to prevent abuse of control messages, since they are used to control the service the subscriber is receiving. The executed phone or data calls are charged to the users, so one could gain financial advantage by forging control messages. In UMTS, this has been handled properly, but GSM lacks decent authentication of control messages.

MAC headers and flow control data seem to be always left unencrypted.

System	Unprotected	Authenticated	Encrypted	Encrypted and authenticated
GSM	Stealing flags		User data Ctrl data	
UMTS	MAC hdrs Flow ctrl		User data	RRC cmds
WLAN	MAC hdrs Flow ctrl MAC cmds	MAC hdrs of data PDUs		SDUs
Bluetooth	Packet hdrs Flow ctrl		User data LMP cmds L2CAP ctrl	
IEEE 802.15.3		MAC hdrs Flow ctrl MAC cmds		SDUs
IEEE 802.15.4		MAC hdrs Flow ctrl		SDUs MAC cmds Beacon pld

Table 6.4: Scope of encryption and authentication

An eavesdropper observing the flow control data can only deduce that data are being sent and see if there is any congestion. This is not very useful information, since transmission of data could be observed even if flow control messages were encrypted. IEEE 802.15.3 and IEEE 802.15.4 authenticate flow control data. Nevertheless, that is not very useful, since it offers protection only against one type of DoS attack that could be carried out much easier by other means.

Since MAC headers are not encrypted in the IEEE systems, there is no protection of location privacy because the MAC addresses are constant and they identify the device. Hence, an eavesdropper having receivers at several locations could track the movements of particular users. In contrast, GSM [ETSI99b] and UMTS [3GPP03e] use temporary user identifiers in order to provide location privacy. However, the location privacy of GSM can be broken by impersonating a base station due to the lack of control message authentication [WW02, Sect. 15.9.5].

Chapter 7

Comparison of Authenticated Encryption Modes

Traditionally, block cipher modes of operation have defined how encryption should be applied when the length of the message is longer than the block size of the cipher. There are five such modes recommended by the US National Institute of Standards and Technology: Electronic Codebook (ECB), Cipher Block Chaining (CBC), Cipher Feedback (CFB), Output Feedback (OFB), and Counter (CTR) [Dwo01]. However, ECB is known to be very insecure and is rarely used in practice [Sti02, Sect. 3.7].

There are also some modes of operation for message authentication, such as CBC-MAC. Recently, many new modes of operation have been proposed that combine functionality for encryption and integrity protection. This chapter discusses and compares six such proposals. Table 7.1 shows the meanings of the symbols used in the following discussion.

7.1 Offset-Based AE Modes

Charanjit Jutla proposed Integrity-Aware Parallelizable Mode (IAPM) [Jut01], providing both encryption and message integrity, and still being fully parallelizable, like ECB and CTR. Offset Codebook (OCB) [RBBK01] is a refinement of IAPM.

Symbol	Meaning
M	Message
A	AAD
N	Nonce
n	Block size of the cipher in bits
$\|X\|$	Length of $X \in \{M, A, N\}$ in bits

Table 7.1: Notation used in discussion on modes of operation

IAPM and OCB resemble the ECB mode, but they generate a random offset for each block that is xored to the input and the output of the cipher, significantly improving security. IAPM suggests two methods for offset generation:

1. Modulo-2 linear combinations of $t = \left\lceil \log_2 \left(\frac{|M|}{n} + 2 \right) \right\rceil$ vectors generated using the block cipher.
2. Using modulo prime addition to generate the offsets from two vectors generated using the block cipher.

OCB uses only an improved version of the modulo-2 generation method, requiring only one evaluation of the block cipher.

OCB has the following advantages over IAPM:

- Message length does not have to be a multiple of the block size of the cipher.
- IVs are not required to be randomly chosen but only unique.[1]
- No different key is required in offset generation.
- Offset generation based on modulo-2 arithmetics is far more efficient.

These differences make OCB more flexible, efficient, and suitable for wireless communications systems.

To authenticate messages, IAPM basically calculates a xor-sum of the plaintext blocks and encrypts it using an offset vector. OCB has some special handling with respect to the final block, but essentially it uses the same mechanism to compute the authentication tag. As noted in Footnote 4 of Chapter 4, this kind of action does not by itself ensure authenticity. However, it has been shown that if the encryption function is assumed to be a strong pseudo-random permutation (SPRP),[2] then authenticity is achieved [BR00, Sect. 4].

7.2 Extensions of CTR Mode

Patents have been filed covering both IAPM and OCB. This has motivated development of patent-free authenticated encryption modes. These modes usually have been extensions of the CTR mode. One such mode was CCM, which has been adopted to several IEEE standards. In fact, OCB appeared in early versions of IEEE 802.11i draft standard but was later replaced by CCM partly due to these intellectual property issues [SE02, Cio02]. CCM combined the CTR mode and CBC-MAC to achieve authenticated encryption. CCM also introduced the possibility to include Additional Authenticated Data (AAD), such as packet headers, to the computation of the authentication tag.[3] CCM is described in more detail in Appendix A.

[1] In [Jut01], two approaches for generating t random vectors were proposed. It seems that only one of them really requires randomly chosen IVs, since the other way involves encrypting the IV before using it as a counter.

[2] The notion of the SPRP property was introduced in [LR88, Sect. 4.4]. Originally, the property was called *super pseudo-randomness* but nowadays term *strong* is more common.

[3] Modes allowing inclusion of such data are sometimes referred to as AE with Associated Data (AEAD) modes. Methods for converting AE modes into AEAD modes exist, and there are such versions of both IAPM [HR03] and OCB [Rog02].

Nevertheless, there were some shortcomings in the CCM mode. Phillip Rogaway (who had filed US Patent application covering OCB [USPt02]) and David Wagner published a paper criticizing several aspects of CCM [RW03]. Later, they contributed to the development of a new mode called EAX [BRW04], offering the following improvements over CCM:

- Word Alignment (WA) is preserved. CCM prepends a few octets of length information to the AAD before applying CBC-MAC, in order to handle AAD with variable length properly. In contrast, EAX uses One-Key CBC-MAC (OMAC), which is by itself secure for arbitrary-length messages [IK03], unlike the standard CBC-MAC [BKR00, Sect. 5].
- EAX is on-line in the sense that the length of the message does not have to be known in advance to start processing.
- Static AAD can be preprocessed, since the processing does not depend on the nonce.[4]
- EAX allows fast verification (without decrypting the message) because OMAC is applied to the ciphertext instead of the plaintext.

EAX did not, however, solve the real problem of CCM: CBC-based authentication is not parallelizable and thus not suitable for high-speed applications. The first parallelizable patent-free AE mode was the Carter–Wegman + CTR (CWC) mode [KVW04]. It uses CTR mode encryption combined to the universal hash approach to message authentication suggested by Mark Wegman and Lawrence Carter [WC81]. However, CWC uses one block of the keystream to mask the output of the hash function instead of a randomly chosen bit string (one-time pad). The CWC hash function is a polynomial function in a finite field, enabling parallel evaluation using Horner's rule. It has been shown to be Almost Xor Universal (AXU) [Kra94, Sect. 2.2],[5] which is a bit weaker notion but still enough to provide integrity in practice.

The most recently proposed mode of operation is the Galois/Counter Mode (GCM) [MV04b]. It is pretty similar to CWC, except it uses polynomial hashing over $GF(2^n)$ instead of $\mathbb{Z}_{2^{127}-1}$,[6] which obviously allows more efficient implementations. Also the GCM hash function family has been proven to be AXU. The authentication codes of CWC and GCM are computed using the ciphertext blocks, like in EAX, thus allowing fast verification.

7.3 Performance

Table 7.2 summarizes the features of the modes discussed above. The following definition applies to the row indicating the number of Block Cipher Evaluations (BCEs) per message:

$$B_X = \left\lceil \frac{|X|}{n} \right\rceil \quad \text{where } X \in \{M, A, N\}$$

[4] In this chapter, *nonce* refers to an IV that is not required to be random but only unique with respect to the key.

[5] In [Kra94], term *otp-secure hash family* was used.

[6] CWC is defined for 128-bit ciphers only and therefore the hashing field is fixed.

In other words, B_X means the length of X in terms of the block size of the cipher. As regards to CCM, $\delta : \mathbb{Z}_{2^{64}} \to \mathbb{Z}_2$ is a function of $|A|$ and is not of interest here, since it does not significantly affect the number of calls. It can be noted that CCM and EAX are very inefficient, since they double the number of block cipher invocations by using CBC-based authentication codes, in addition to losing parallelizability.

GCM seems to use the lowest possible number of BCEs per message: one evaluation per each message block and one for masking the output of the universal hash function. CWC applies the block cipher also to the output of the hash function itself before masking. The aim is purportedly to distribute the output of the hash function uniformly over $\mathbb{Z}_{2^{128}}$. Otherwise, values belonging to $\mathbb{Z}_{2^{128}} \setminus \mathbb{Z}_{2^{127}-1}$ would never occur. In particular, since the masking operation is addition in $\mathbb{Z}_2^{128}$, the MSB of the 128-bit authentication code would not depend on the message at all but only on the key and the nonce, thus giving a forgery advantage to an adversary. This difference of one BCE may seem indifferent but it is not. In pipelined hardware implementations, a stall is caused by this extra evaluation because it depends on the results of the B_M previous evaluations [MV04a, Sect. 3.1].

It is clear that OCB is more efficient than IAPM and GCM is faster than CWC. But how do OCB and GCM relate to each other in terms of performance? In hardware implementations, the same pipeline stall argument holds in favor of GCM: In GCM, all BCEs are independent of each other, whereas OCB causes even two pipeline stalls per message.[7] As regards to software implementations, GCM outperformed OCB with respect to short messages, according to simulations by the authors of GCM. However, OCB was faster when encrypting long messages. This observation is explained so that GCM is faster due to using one BCE less than OCB, but computing the hash function is more expensive than simple modulo-2 addition, so with long messages OCB comes off better. [MV04a, Sect. 3]

It should be noted, however, that GCM implementations used in the tests cited above used time–memory tradeoffs to speed up computing the hash function, as recommended by the GCM specifications. The specifications even admit that if no key-dependent memory is used, the hash function computation generally takes even more time than running AES [MV04b, Sect. 4.1]. Therefore, one cannot conclude that GCM is unequivocally faster than OCB in encryption of short messages.

7.4 Security

All the AE modes discussed in this chapter have been proven to be secure. More precisely, they satisfy *indistinguishability under chosen ciphertext attack* (IND-CCA) [RS91, Sect. 2] and *integrity of ciphertexts* (INT-CTXT) [BN00, Sect. 2] notions of security. The proofs assume that the encryption function is a pseudo-random permutation (PRP), leveraging the result that PRPs can be

[7] As can be seen in Table 7.2, IAPM has less pipeline stalls per message than OCB. However, both stalls in OCB are due to fixing shortcomings of IAPM: requiring random IVs and disallowing short final blocks.

Mode	IAPM	OCB	CCM	EAX	CWC	GCM
General						
Message length	Multiple of block size	Arbitrary	Multiple of 8 bits	Arbitrary	Multiple of 8 bits	Arbitrary
IV requirements	Random	Unique	Unique	Unique	Unique	Unique
IV size	Block size	Block size	56–104 bits	Arbitrary	80 bits	Arbitrary
Key count	2	1	1	1	1	1
IPR issues	Yes	Yes	No	No	No	No
Overall efficiency						
BCEs/message	(1) $B_M + 2 + t$ (2) $B_M + 4$	$B_M + 2$	$2B_M + B_A + 2 + \delta$	$2B_M + B_A + B_N$	$B_M + 2$	$B_M + 1$
BCEs/session setup	0	1	0	0	1	1
Parallelizable	Yes	Yes	No	No	Yes	Yes
Pipeline stalls/msg	0–1	2	—	—	1	0
WA preservation	Yes	Yes	No	Yes	Yes	Yes
On-line	Yes	Yes	No	Yes	Yes	Yes
Encryption						
Ciphering method	Offset	Offset	Stream (CTR)	Stream (CTR)	Stream (CTR)	Stream (CTR)
Offset generation modulus	(1) 2 (2) Prime	2	—	—	—	—
Authentication						
Code type	Encrypted xor sum	Encrypted xor sum	CBC-MAC	OMAC	Universal hash over $\mathbb{Z}_{2^{127}-1}$	Universal hash over $GF(2^n)$
AAD	No	No	Yes	Yes	Yes	Yes
Static AAD precomputation	—	—	No	Yes	Yes	Yes
Fast verification	No	No	No	Yes	Yes	Yes

Table 7.2: Features of AE modes of operation

used as pseudo-random functions [GGM86, BKR00].[8]

However, to achieve authenticity with IAPM and OCB, one must make a stronger assumption about the encryption function. As pointed out in Section 7.1, it is not possible to ensure authenticity by inducing public redundancy before encryption unless the encryption function is an SPRP. Therefore, IAPM and OCB might be considered somewhat weaker than the other modes.

The security proofs can be found in the papers cited in Sections 7.1 and 7.2, except for GCM, the proof for which is given in [MV04a]. The OCB security proof is presented in more detail in [RBB03], and the proof for CCM is presented in [Jon02]. Note that these papers prove only indistinguishability under chosen *plaintext* attack (IND-CPA) [GM84, Sect. 5.1], which is a weaker notion than IND-CCA [DDN00]. However, it has been shown that IND-CPA and INT-CTXT together imply IND-CCA [BN00, Sect. 3].

7.4.1 Attacks on AE Modes

Niels Ferguson, one of the authors of CCM, presented a collision attack on OCB. The attack exploits the fact that in OCB, the xor sum of any two plaintext or ciphertext blocks, excluding the final blocks, is a function of only the key. In other words, the sum is independent of the nonce. Hence, observing two such sums that are equal means that the block cipher inputs are equal with a significant probability (0.5). The attack does not contradict with the security proof of OCB, thus requiring huge amounts of known plaintext while the forgery probability still remains pretty low. However, the paper showed that the number of blocks encrypted using a single key (B) should be limited to $2^{\frac{n-\tau}{2}}$, where τ is the length of the authentication tag in bits, because the effective length of the tag is limited to $n - 2\log_2 B$. Otherwise it is no use having a longer tag. [Fer02]

CCM, EAX, CWC, and GCM were extensions of CTR mode. Therefore, examining the security of CTR may help drawing conclusions about those modes. It has been shown that when using CTR, B should be limited to about $2^{\frac{n}{2}}$ [BDJR97]. The reason is that if counter values are not reused (which is the right thing to do), there will be no identical keystream blocks. However, a purely random string would contain identical blocks with significant probability if it was $2^{\frac{n}{2}} \cdot n$ bits long, due to the birthday paradox, thus giving the attacker a significant distinguishing advantage between the encryption output and random strings. [McG02, Sect. 3]

This can be seen when substituting $q = B$ and $M = 2^n$ in Equation (6.1):

$$B \approx \sqrt{2 \ln \frac{1}{1-\epsilon}} \cdot 2^{\frac{n}{2}} \tag{7.1}$$

Assume that the limit for B is set strictly to $2^{\frac{n}{2}}$. Solving

$$\sqrt{2 \ln \frac{1}{1-\epsilon}} = 1 \tag{7.2}$$

for ϵ yields a probability of 0.39 for occurrence of two identical blocks in a random string, which clearly is significant. However, attacks based on this

[8] The original definition of the PRP notion was given in [LR88, Sect. 4.3]. In [BKR00], a bit different definition is used, modeling better real block ciphers.

property of CTR mode are not very practical, since they have enormous memory requirements due to the necessity of remembering which keystream blocks have been used.

CTR is also known to be prone to precomputation attacks. An attacker who is able to predict a counter value that is likely to be used in future, computes a keystream block for with many possible key values. On observing a block encrypted under that counter value, he can check which keystream block and key have been used, assuming that he knows the corresponding plaintext block. Indeed, this attack requires huge amounts of memory and preprocessing. Anyway, one possible defense is to include an unpredictable or at least uniformly distributed part to the initial counter value. [McG02, Sect. 4]

It is interesting that in EAX, the nonce is always jumbled with OMAC before it is used, thus rendering the actual counter values unpredictable. In GCM, the nonce is hashed in the case where $|N| \neq n - 32$, but otherwise it is used directly to initialize the counter.[9] CCM nor CWC provide no randomization to the nonce. Therefore, EAX might be considered more secure against precomputation attacks.

[9] $n - 32$ bits is the recommended nonce size for high-speed implementations for the reason that then hashing is omitted.

Chapter 8

Conclusions

This thesis thoroughly analyzed the details of security features of several radio communications systems, in order to help the reader gain profound understanding of them and to design new systems. While analyzing those details, theories were developed on certain aspects of security functions. Session key exchange procedures were examined, and the results are reviewed in Section 8.1. A theory on useful inputs to data protection functions was devised and it is summarized in Section 8.2, whereas Section 8.3 concerns itself with replay prevention in the case of burst transmissions.

In addition to these theoretical results, a few results concerning specific systems, GSM, Bluetooth, and IEEE 802.15.3, were gained. These are reviewed in Section 8.4. Moreover, six AE modes of operation were examined, and that discussion is reviewed in Section 8.5.

8.1 Session Key Establishment

In almost all systems we studied, security associations were characterized by a long-term link key and shorter-lived session keys that were used with the actual cryptographic algorithms. The link keys were used to set up session keys such that an eavesdropper cannot deduce them, having seen the related messages. The primary means for this were to

- derive session keys from the link key using one-way functions and randomly generated numbers, or
- encrypt transmission of randomly generated session keys using the link key.

Indeed, there are other well-known secure key exchange mechanisms based on public key cryptography, such as the Station-to-Station protocol [DOW92, Sect. 5] based on the Diffie–Hellman key exchange [DH76, Sect. 3]. However, public key methods are not usually leveraged in radio communications systems, at least yet. More information about key exchange protocols can be found in [BM03].

As the examples of GSM and Bluetooth showed, it is extremely important to safeguard against key replay attacks. The following methods for this were identified:

- Each party going to encrypt data or verify authenticity of data using a new key is given a means to affect it. The link key should affect the session key derivation in an unpredictable manner.

- Each party going to encrypt data or verify authenticity of data using a new key is given a means to challenge the contributing parties to prove that they know the link key.

- Key agreement procedure is protected by sequence numbers. Link key-based message authentication codes are used to verify the messages and their sequence numbers.

Recall that using synchronized clocks as sequence numbers requires undertaking additional counter-measures, as stressed by the example in Section 6.1.1.

8.2 Specifying Requisite Input Values

In addition to plaintext and the session key, encryption and integrity protection functions need some supplementary inputs too. Some systems use proprietary stream ciphers and authentication functions, in which case the functions have been specified to have appropriate input arguments. Other systems use AES in CCM mode and embed the additional inputs to the nonce required by that mode. We saw that there were basically three different classes of inputs:

Counter (CTR) that is incremented after one frame or fragment thereof has been sent, or when another counter rolls over. Alternatively, counters may be based on synchronized clocks.

Counter Identifier (CID) that combined with the other inputs of this type uniquely identifies the set of counters, the values of which have been used as input.

Counter Skew Guard (CSG) that provides protection against replay attacks when one communicating party is able to unilaterally reinitialize counters during the lifetime of the session key.

On defining an encryption and integrity protection protocol for a specific system, it should be determined what kind of counters are used. If the clocks of the communicating parties are synchronized, they could be directly used as the counter. However, there might be reasons not to do so, such as:

- Contention-based multiple access scheme necessitates using sequence numbers such that each transmitter has its own counter.

- Layer independence and cohesion can be increased by maintaining separate counters at the link layer.

- Using session key-specific packet sequence numbers helps determining when it is necessary to renew the key.

- *En route* packet reordering according to priority requires the receiver to maintain a distinct counter for each priority class.

Having defined the counters, such set of CIDs should be determined that uniquely identifies the set of the counters that are used as input. Moreover, if handovers are possible in the system, including CSG-type inputs should be considered, according to the rules given in Section 6.3.4. Using synchronized clocks as sequence numbers usually requires using CSGs too.

8.3 Burst Transmissions

Burst transmissions used in high-rate radios impose challenges to message replay prevention. Defining a simple replay counter and requiring packet numbers to be monotonically increasing does not work if it is allowed to request retransmission of solitary packets. Three main methods for handling the situation properly were presented:

- Renumber and re-encrypt packets on retransmission.

- If the packet order is preserved by the link layer, apply replay filtering after order restoration.

- Use the RDSR algorithm described in Section 6.4.1.

8.4 Shortcomings in Security Specifications

Some security vulnerabilities were detected in GSM, IEEE 802.15.3, and Bluetooth, and they are reviewed in this section. It is interesting that in all these three systems, an active attacker can

- set up a previously used encryption key, and

- rewind the clock (or counter) in order to replay recorded packets and force keystream reuse.

8.4.1 GSM

As pointed out by Barkan *et al.*, the GSM system is vulnerable to key replay attacks. They also showed how that vulnerability can be exploited. Their attack is very realistic, as it requires only acquiring a few dozen milliseconds of ciphertext by impersonating a base station.

When analyzing the inputs to the encryption functions, it was perceived that there were no CSG-type inputs to the GSM encryption algorithm, although handover is possible. Therefore, a malicious base station could force the MS to reuse keystreams, since it is not required to change the ciphering key after handovers. Note that although not very probable, making the MS reuse keystreams can happen accidentally if the clock of the new base station happens to be slightly arrears with that of the old station. In this case, even a passive eavesdropper could exploit this vulnerability.

8.4.2 IEEE 802.15.3

We saw that in IEEE 802.15.3, the replay prevention mechanism is deficient. It allows replaying frames within a single superframe, provided that the SDU number finds time to wrap around. This problem could still be fixed by requiring an SFC replay counter for each peer device, since packets are re-encrypted before retransmission. Also the other solutions proposed in Section 6.4 would be compliant with the specification, except for the one suggesting combining the SDU number and the SFC.

As PNC-supplied time tokens are used as sequence numbers, other devices are prone to key replay attacks if they have lost synchronization with the PNC, possibly due to visiting another piconet. This shortcoming can also be exploited to replay other previously recorded frames too, and to collect ciphertexts encrypted using the same keystream. A CSG-type input to the CCM nonce would have removed this vulnerability.

8.4.3 Bluetooth

Note: Parts of this section will be separately published in [RN04].

In Section 4.3, it was demonstrated that just introducing a second stronger encryption algorithm is not sufficient to upgrade the security level of Bluetooth encryption as desired. The root cause of the problem is that it may be possible for a master device to replay authentication and key exchange, thus making the attack by Barkan *et al.* conceptually applicable on Bluetooth too.

Indeed, the key replay problem had already been made evident by Gauthier. However, this thesis exhibited the problem arising from ACO reuse, which makes the replay attack even easier. Moreover, the solutions suggested in [3GPP04k] were quite specific to the EAP-AKA protocol. In this thesis, it was demonstrated that the problem is more general and requires more general counter-measures too.

In Section 4.3.4, four alternative approaches to protect against the adapted Barkan–Biham–Keller attack were proposed:

- The link key is changed frequently.
- Bluetooth application profiles mandate the slave to provide the last authentication challenge before encryption key derivation *and* forbid using a single ACO to derive several encryption keys.
- Encryption algorithm negotiation is authenticated.
- Different key derivation algorithms are specified for each encryption algorithm.

The first is contradictory to the idea of link keys. According to the specifications, combination keys are semi-permanent, although changing them frequently would really increase security against active attacks. The second approach is neither in line with the specifications, which tell the implementors to store the ACO for future use. The specifications should rather encourage avoiding ACO reuse under all circumstances.

The last two approaches would only thwart the attacks on multiple encryption algorithms presented in this paper. The key replay attack by Gauthier would still remain valid. Moreover, they would not prevent the clock skew attack either. An attacker impersonating the master could rewind the clock when the slave is temporarily absent, and then replay the key and force keystream reuse. But either one of the first two counter-measures would also work against these attacks too, and therefore are the recommended options.

There is also another weakness in Bluetooth encryption. There are three encryption modes, one of which allows encryption of broadcast messages. However, in that mode, unicast messages are also encrypted using the common key, thus allowing eavesdropping by other piconet members. Therefore, fourth encryption mode was proposed that would solve this problem. In that mode, encryption of unicast messages would use point-to-point encryption keys, whereas broadcast messages would be protected using the same group key used by the devices not supporting this new mode. Hence, this solution would work in spite of possible legacy or limited-capability devices belonging to the piconet.

8.5 AE Modes

Six different AE-type modes of operation for block ciphers were compared: IAPM, OCB, CCM, EAX, CWC, and GCM. All these modes have been proven to be IND-CCA, assuming the block cipher to be a PRP, or SPRP for IAPM and OCB. Some collision and precomputation attacks were discussed but they require huge amounts of memory to succeed with a non-negligible probability. Although differences in resistance to these attacks exist between the modes, they all can be considered reasonably secure as of this writing.

GCM appears to have most impressive characteristics of these modes. It preserves WA, is on-line in the sense that the message size does not have to be known in advance, is fully parallelizable, and allows AAD preprocessing and fast verification. As regards to performance, it uses the optimal number of BCEs without causing pipeline stalls in hardware implementations. The only dubious property is its use of a polynomial hash function. Although parallelizable using Horner's rule, it has pathetic evaluation time when compared to OCB's authentication method. Therefore, OCB suits better long messages, if performance is an issue. Moreover, to keep GCM evaluation times bearable even with shorter messages, a time–memory tradeoff must be used.

Bibliography

[3GPP02a] 3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; 3G Security; Specification of the 3GPP Confidentiality and Integrity Algorithms; Document 1: *f8* and *f9* Specification (Release 5). 3GPP TS 35.201 V5.0.0, June 2002.

[3GPP02b] 3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; 3G Security; Specification of the 3GPP Confidentiality and Integrity Algorithms; Document 2: KASUMI Specification (Release 5). 3GPP TS 35.202 V5.0.0, June 2002.

[3GPP03a] 3rd Generation Partnership Project; Technical Specification Group Radio Access Network; Radio Interface Protocol Architecture (Release 6). 3GPP TS 25.301 V6.0.0, December 2003.

[3GPP03b] 3rd Generation Partnership Project; Technical Specification Group Radio Access Network; Radio Link Control (RLC) protocol specification (Release 6). 3GPP TS 25.322 V6.0.0, December 2003.

[3GPP03c] 3rd Generation Partnership Project; Technical Specification Group Radio Access Network; Packet Data Convergence Protocol (PDCP) Specification (Release 6). 3GPP TS 25.323 V6.0.0, December 2003.

[3GPP03d] 3rd Generation Partnership Project; Technical Specification Group Radio Access Network; UTRAN overall description (Release 6). 3GPP TS 25.401 V6.2.0, December 2003.

[3GPP03e] 3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; 3G Security; Security architecture (Release 6). 3GPP TS 33.102 V6.0.0, September 2003.

[3GPP04a] 3rd Generation Partnership Project; Technical Specification Group Services and Systems Aspects; Vocabulary for 3GPP Specifications (Release 6). 3GPP TS 21.905 V6.6.0, March 2004.

[3GPP04b] 3rd Generation Partnership Project; Technical Specification Group Services and Systems Aspects; Network architecture (Release 6). 3GPP TS 23.002 V6.4.0, March 2004.

[3GPP04c] 3rd Generation Partnership Project; Technical Specification Group Services and Systems Aspects; General Packet Radio Service (GPRS); Service description; Stage 2 (Release 6). 3GPP TS 23.060 V6.4.0, March 2004.

[3GPP04d] 3rd Generation Partnership Project; Technical Specification Group Radio Access Network; Physical channels and mapping of transport channels onto physical channels (FDD) (Release 6). 3GPP TS 25.211 V6.1.0, June 2004.

[3GPP04e] 3rd Generation Partnership Project; Technical Specification Group Radio Access Network; Physical channels and mapping of transport channels onto physical channels (TDD) (Release 6). 3GPP TS 25.221 V6.1.0, June 2004.

[3GPP04f] 3rd Generation Partnership Project; Technical Specification Group Radio Access Network; Medium Access Control (MAC) protocol specification (Release 6). 3GPP TS 25.321 V6.1.0, March 2004.

[3GPP04g] 3rd Generation Partnership Project; Technical Specification Group Radio Access Network; Broadcast/Multicast Control (BMC) (Release 6). 3GPP TS 25.324 V6.1.0, June 2004.

[3GPP04h] 3rd Generation Partnership Project; Technical Specification Group Radio Access Network; Radio Resource Control (RRC); Protocol Specification (Release 6). 3GPP TS 25.331 V6.1.0, March 2004.

[3GPP04i] 3rd Generation Partnership Project; Technical Specification Group Core Network; Mobile Application Part (MAP) specification (Release 6). 3GPP TS 29.002 V6.5.0, March 2004.

[3GPP04j] 3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; 3G Security; Cryptographic algorithm requirements (Release 6). 3GPP TS 33.105 V6.0.0, June 2004.

[3GPP04k] Notes on Gauthier's replay attack on the UE functionality split scenario. 3GPP TSG SA WG3 Meeting #32, S3-040091, February 2004.

[AB01] J. H. An and M. Bellare. Does encryption with redundancy provide authenticity? In Pfitzmann [Pfi01], pages 512–528.

[ABV$^+$04] B. Aboba, L. Blunk, J. Vollbrecht, J. Carlson, and H. Levkowetz. Extensible Authentication Protocol (EAP). RFC 3748, IETF Network Working Group, June 2004.

[AH04] J. Arkko and H. Haverinen. Extensible Authentication Protocol Method for UMTS Authentication and Key Agreement (EAP-AKA). Internet Draft (`draft-arkko-pppext-eap-aka-12.txt`), April 2004.

[AK03] F. Armknecht and M. Krause. Algebraic attacks on combiners with memory. In Boneh [Bon03], pages 162–176.

[ALP04] F. Armknecht, J. Lano, and B. Preneel. Extending the framework of the resynchronization attack. In *Proceedings of Selected Areas in Cryptography 2004*, Lecture Notes in Computer Science, Waterloo, Ontario, Canada, August 2004. University of Waterloo, Springer.

[Arm04a] F. Armknecht. Algebraic attacks on stream ciphers. Presentation in minisymposium "Secure Crypto for Industry" at ECCOMAS 2004, July 2004.

[Arm04b] F. Armknecht. Improving fast algebraic attacks. In Roy and Meier [RM04], pages 65–82.

[BBK03] E. Barkan, E. Biham, and N. Keller. Instant ciphertext-only cryptanalysis of GSM encrypted communications. In Boneh [Bon03], pages 600–616.

[BDJR97] M. Bellare, A. Desai, E. Jokipii, and P. Rogaway. A concrete security treatment of symmetric encryption. In *Proceedings of FOCS '97*, pages 394–403, Miami Beach, FL, USA, October 1997. IEEE Computer Society.

[BGW99] M. Briceno, I. Goldberg, and D. Wagner. A pedagogical implementation of the GSM A5/1 and A5/2 "voice privacy" encryption algorithms, 1999. [referenced on June 28, 2004] `http://www.mirrors.wiretapped.net/security/cryptography/algorithms/gsm/a5-1-2.c`.

[BGW01] N. Borisov, I. Goldberg, and D. Wagner. Intercepting mobile communications: The insecurity of 802.11. In *Proceedings of MOBICOM 2001*, pages 180–189, Rome, Italy, July 2001. Association for Computing Machinery.

[BKR00] M. Bellare, J. Kilian, and P. Rogaway. The security of the Cipher Block Chaining Message Authentication Code. *Journal of Computer and System Sciences*, 61(3):362–399, December 2000.

[Blue03a] Architecture & Terminology Overview. Volume 1 of Specification of the Bluetooth System, Version 1.2. Promoter Members of Bluetooth SIG, November 2003.

[Blue03b] Core System Package [Controller volume]. Volume 2 of Specification of the Bluetooth System, Version 1.2. Promoter Members of Bluetooth SIG, November 2003.

[BM03] C. Boyd and A. Mathuria. *Protocols for Authentication and Key Establishment.* Springer, Berlin, Germany, June 2003.

[BN00] M. Bellare and C. Namprempre. Authenticated encryption: Relations among notions and analysis of the generic composition paradigm. In Okamoto [Oka00], pages 531–545.

[Bon03] D. Boneh, editor. *Advances in Cryptology — CRYPTO 2003, 23rd Annual International Cryptology Conference, Proceedings*, volume 2729 of *Lecture Notes in Computer Science*, Santa Barbara, CA, USA, August 2003. Springer.

[BR00] M. Bellare and P. Rogaway. Encode-then-encipher encryption: How to exploit nonces or redundancy in plaintexts for efficient cryptography. In Okamoto [Oka00], pages 317–330.

[Bro04] C. Brookson. A5/2 withdrawal from handsets. 3GPP TSG SA WG3 Meeting #33, S3-040376, May 2004.

[BRW04] M. Bellare, P. Rogaway, and D. Wagner. The EAX mode of operation. In Roy and Meier [RM04], pages 389–407.

[BSW01] A. Biryukov, A. Shamir, and D. Wagner. Real time cryptanalysis of A5/1 on a PC. In B. Schneier, editor, *Proceedings of Fast Software Encryption 2000*, volume 1978 of *Lecture Notes in Computer Science*, pages 1–18, New York, NY, USA, April 2001. Springer.

[Cio02] F. Ciotti. Minutes of IEEE P802.11i plenary session, Vancouver, British Columbia, Canada, July 2002.

[Cou03] N. T. Courtois. Fast algebraic attacks on stream ciphers with linear feedback. In Boneh [Bon03], pages 177–194.

[DDN00] D. Dolev, C. Dwork, and M. Naor. Nonmalleable cryptography. *SIAM Journal on Computing*, 30(2):391–437, April 2000.

[DH76] W. Diffie and M. E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22(6):644–654, November 1976.

[DH98] S. Deering and R. Hinden. Internet Protocol, Version 6 (IPv6) Specification. RFC 2460, IETF Network Working Group, December 1998.

[Dom02] A. Doman. *Essential Guide to Wireless Communications Applications: From Cellular Systems to Wi-Fi.* Prentice Hall PTR, Upper Saddle River, NJ, USA, 2nd edition, May 2002.

[DOW92] W. Diffie, P. C. van Oorschot, and M. J. Wiener. Authentication and authenticated key exchanges. *Designs, Codes and Cryptography*, 2(2):107–125, June 1992.

[Dwo01] M. Dworkin. Recommendation for Block Cipher Modes of Operation: Methods and Techniques. NIST Special Publication 800-38A, 2001 Edition, December 2001.

[Dwo04] M. Dworkin. Recommendation for Block Cipher Modes of Operation: The CCM Mode for Authentication and Confidentiality. NIST Special Publication 800-38C, May 2004.

[EJ00] P. Ekdahl and T. Johansson. Some results on correlations in the Bluetooth stream cipher. In *Proceedings of the 10th Joint Conference on Communications and Coding*, Obertauern, Austria, 2000.

[ETSI99a] Digital cellular telecommunications system (Phase 2+); Mobile Stations (MS) features. GSM TS 02.07 version 8.0.0 Release 1999, July 1999.

[ETSI99b] Digital cellular telecommunications system (Phase 2+); Security related network functions. ETSI GSM TS 03.20 version 8.0.0 Release 1999, November 1999.

[ETSI00a] Digital cellular telecommunications system (Phase 2+); Mobile radio interface layer 3 specification. ETSI GSM TS 04.08 version 7.8.0 Release 1998, June 2000.

[ETSI00b] Digital cellular telecommunications system (Phase 2+); Physical layer on the radio path; General description. ETSI GSM TS 05.01 version 8.4.0 Release 1999, July 2000.

[ETSI00c] Digital cellular telecommunications system (Phase 2+); Multiplexing and multiple access on the radio path. ETSI GSM TS 05.02 version 8.5.1 Release 1999, November 2000.

[ETSI00d] Digital cellular telecommunications system (Phase 2+); Channel coding. ETSI GSM TS 05.03 version 8.5.0 Release 1999, July 2000.

[ETSI00e] Digital cellular telecommunications system (Phase 2+); Mobile Application Part (MAP) specification. ETSI GSM TS 09.02 version 7.5.1 Release 1998, June 2000.

[Fer02] N. Ferguson. Collision attacks on OCB. NIST Public Comments on Modes of Operation, February 2002.

[Gau04] E. Gauthier. A man-in-the-middle attack using Bluetooth in a WLAN interworking environment. 3GPP TSG SA WG3 Meeting #32, S3-040163, February 2004.

[GGM86] O. Goldreich, S. Goldwasser, and S. Micali. How to construct random functions. *Journal of the ACM*, 33(4):792–807, October 1986.

[GM84] S. Goldwasser and S. Micali. Probabilistic encryption. *Journal of Computer and System Sciences*, 28(2):270–299, April 1984.

[GSMA04] Membership & market statistics as at the end of February 2004. GSM Association, 2004. [referenced on May 28, 2004] `http://www.gsmworld.com/news/statistics/pdf/feb04.pdf`.

[HN99] M. Hermelin and K. Nyberg. Correlation properties of the Bluetooth combiner. In J-S. Song, editor, *Proceedings of ICISC '99*, volume 1787 of *Lecture Notes in Computer Science*, pages 17–29, Seoul, South Korea, December 1999. Korea University, Springer.

[HR03] P. Hawkes and G. G. Rose. A mode of operation with Partial Encryption and Message Integrity (PEMI). IACR Cryptology ePrint Archive, Report 2003/001, January 2003.

[HS04] H. Haverinen and J. Salowey. Extensible Authentication Protocol Method for GSM Subscriber Identity Modules (EAP-SIM). Internet Draft (`draft-haverinen-pppext-eap-sim-13.txt`), April 2004.

[HT04] H. Holma and A. Toskala, editors. *WCDMA for UMTS: Radio Access for Third Generation Mobile Communications.* John Wiley & Sons, Chichester, England, UK, 3rd edition, July 2004.

[IEEE97] Information Technology; Telecommunications and information exchange between systems; Local and metropolitan area networks; Specific requirements; Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. ANSI/IEEE Standard 802.11, July 1997.

[IEEE99] Information Technology; Telecommunications and information exchange between systems; Local and metropolitan area networks; Specific requirements; Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. ANSI/IEEE Standard 802.11, 1999 Edition, March 1999.

[IEEE01] IEEE Standard for Local and metropolitan area networks; Port-Based Network Access Control. IEEE Standard 802.1X-2001, July 2001.

[IEEE02] IEEE Standard for Information Technology; Telecommunications and information exchange between systems; Local and metropolitan area networks; Specific requirements; Part 15.1: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Wireless Personal Area Networks (WPANs). IEEE Standard 802.15.1-2002, June 2002.

[IEEE03a] IEEE Standard for Information Technology; Telecommunications and information exchange between systems; Local and metropolitan area networks; Specific requirements; Part 15.3: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for High Rate Wireless Personal Area Networks (WPANs). IEEE Standard 802.15.3-2003, September 2003.

[IEEE03b] IEEE Standard for Information Technology; Telecommunications and information exchange between systems; Local and metropolitan area networks; Specific requirements; Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs). IEEE Standard 802.15.4-2003, October 2003.

[IEEE04] IEEE Standard for Information Technology; Telecommunications and information exchange between systems; Local and metropolitan area networks; Specific requirements; Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications; Amendment 6: Medium Access Control (MAC) Security Enhancements. IEEE Standard 802.11i-2004, July 2004.

[IK03] T. Iwata and K. Kurosawa. OMAC: One-key CBC-MAC. In T. Johansson, editor, *Proceedings of Fast Software Encryption 2003*, volume 2887 of *Lecture Notes in Computer Science*, pages 129–153, Lund, Sweden, February 2003. Springer.

[ISO99] Information technology — Security techniques — Message Authentication Codes (MACs) — Part 1: Mechanisms using a block cipher. ISO/IEC 9797-1:1999, 1999.

[Jon02] J. Jonsson. On the security of CTR + CBC-MAC. In K. Nyberg and H. M. Heys, editors, *Proceedings of Selected Areas in Cryptography 2002*, volume 2595 of *Lecture Notes in Computer Science*, pages 76–93, St. John's, Newfoundland, Canada, August 2002. Springer.

[Jut01] C. S. Jutla. Encryption modes with almost free message integrity. In Pfitzmann [Pfi01], pages 529–544.

[JW01] M. Jakobsson and S. Wetzel. Security weaknesses in Bluetooth. In D. Naccache, editor, *Proceedings of the Cryptographer's Track at RSA Conference 2001*, volume 2020 of *Lecture Notes in Computer Science*, pages 176–191, San Francisco, CA, USA, April 2001. RSA Security, Inc., Springer.

[KBC97] H. Krawczyk, M. Bellare, and R. Canetti. HMAC: Keyed-Hashing for Message Authentication. RFC 2104, IETF Network Working Group, February 1997.

[Kra94] H. Krawczyk. LFSR-based hashing and authentication. In Y. Desmedt, editor, *Advances in Cryptology — CRYPTO '94*, volume 839 of *Lecture Notes in Computer Science*, pages 129–139, Santa Barbara, CA, USA, August 1994. Springer.

[Kra02] M. Krause. BDD-based cryptanalysis of key stream generators. In L. R. Knudsen, editor, *Advances in Cryptology — EUROCRYPT 2002*, volume 2332 of *Lecture Notes in Computer Science*, pages 222–237, Amsterdam, The Netherlands, 2002. Springer.

[KVW04] T. Kohno, J. Viega, and D. Whiting. CWC: A high-performance conventional authenticated encryption mode. In Roy and Meier [RM04], pages 408–426.

[LR88] M. Luby and C. Rackoff. How to construct pseudorandom permutations from pseudorandom functions. *SIAM Journal on Computing*, 17(2):373–386, April 1988.

[MAGN] About MAGNET. IST Project 507102. [referenced on September 1, 2004]
`http://www.ist-magnet.org/objectives.html`.

[McG02] D. A. McGrew. Counter mode security: Analysis and recommendations, November 2002. [referenced on August 25, 2004]
`http://www.mindspring.com/~dmcgrew/ctr-security.pdf`.

[MKK98] J. L. Massey, G. H. Khachatrian, and M. K. Kuregian. Nomination of SAFER+ as Candidate Algorithm for the Advanced Encryption Standard (AES). 1st AES Conference, Ventura, CA, USA, August 1998.

[MOV96] A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone. *Handbook of Applied Cryptography.* CRC Press, Boca Raton, FL, USA, October 1996.

[MP92] M. Mouly and M-B. Pautet. *The GSM System for Mobile Communications*. Published by the authors, 1992.

[MV04a] D. A. McGrew and J. Viega. The security and performance of the Galois/Counter Mode (GCM) of operation. IACR Cryptology ePrint Archive, Report 2004/193, August 2004.

[MV04b] D. A. McGrew and J. Viega. The Galois/Counter Mode of Operation (GCM). Submission to NIST Modes of Operation Process, January 2004.

[Nie04] N. Niebert. Ambient Networks Project Overview and Dissemination Plan. IST Project 507134, Deliverable 1.1, Version 1.0, June 2004.

[NIST95] Secure Hash Standard. NIST FIPS Publication 180-1, April 1995.

[NIST01] Specification for the Advanced Encryption Standard (AES). NIST FIPS Publication 197, November 2001.

[NN03] V. Niemi and K. Nyberg. *UMTS Security*. John Wiley & Sons, Chichester, England, UK, 2003.

[OICT03] Information Security Guideline for NSW Government — Part 3: Information Security Baseline Controls, Issue 3.0. New South Wales Office of Information and Communications Technology, June 2003.

[Oka00] T. Okamoto, editor. *Advances in Cryptology — ASIACRYPT 2000, 6th International Conference on the Theory and Application of Cryptology and Information Security, Proceedings*, volume 1976 of *Lecture Notes in Computer Science*, Kyoto, Japan, December 2000. Springer.

[Pfi01] B. Pfitzmann, editor. *Advances in Cryptology — EUROCRYPT 2001, International Conference on the Theory and Application of Cryptographic Techniques, Proceedings*, volume 2045 of *Lecture Notes in Computer Science*, Innsbruck, Austria, May 2001. Springer.

[PkL03] Laki pakkokeinolain muuttamisesta. The Finnish Legislation, Act Number 646/2003, June 2003.

[Pos80] J. Postel. User Datagram Protocol. RFC 768, August 1980.

[Pos81a] J. Postel. Internet Protocol. RFC 791, DARPA Internet Program, Protocol Specification, September 1981.

[Pos81b] J. Postel. Transmission Control Protocol. RFC 793, DARPA Internet Program, Protocol Specification, September 1981.

[RBB03] P. Rogaway, M. Bellare, and J. Black. OCB: A block-cipher mode of operation for efficient authenticated encryption. *ACM Transactions on Information and System Security*, 6(3):365–403, August 2003.

[RBBK01] P. Rogaway, M. Bellare, J. Black, and T. Krovetz. OCB: A block-cipher mode of operation for efficient authenticated encryption. In *Proceedings of ACM CCS 2001*, pages 196–205, Philadelphia, PA, USA, November 2001. Association for Computing Machinery.

[Riv92] R. Rivest. The MD5 Message-Digest Algorithm. RFC 1321, IETF Network Working Group, April 1992.

[RM04] B. K. Roy and W. Meier, editors. *Fast Software Encryption, 11th International Workshop, FSE 2004, Revised Papers*, volume 3017 of *Lecture Notes in Computer Science*, Delhi, India, February 2004. Springer.

[RN04] K. Ritvanen and K. Nyberg. Upgrade of Bluetooth encryption and key replay attack. In *Proceedings of NORDSEC 2004*, Publications in Telecommunications Software and Multimedia, Espoo, Finland, November 2004. Department of Computer Science and Engineering, Helsinki University of Technology.

[Rog02] P. Rogaway. Authenticated-encryption with associated-data. In V. Atluri, editor, *Proceedings of ACM CCS 2002*, pages 98–107, Washington, DC, USA, November 2002. Association for Computing Machinery.

[RS91] C. Rackoff and D. Simon. Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In J. Feigenbaum, editor, *Advances in Cryptology — CRYPTO '91*, volume 576 of *Lecture Notes in Computer Science*, pages 433–444, Santa Barbara, CA, USA, August 1991. Springer.

[Rue85] R. A. Rueppel. Correlation immunity and the summation generator. In H. C. Williams, editor, *Advances in Cryptology — CRYPTO '85*, volume 218 of *Lecture Notes in Computer Science*, pages 260–272, Santa Barbara, CA, USA, August 1985. Springer.

[RW03] P. Rogaway and D. Wagner. A critique of CCM. IACR Cryptology ePrint Archive, Report 2003/070, February 2003.

[SCFJ03] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson. RTP: A Transport Protocol for Real-Time Applications. RFC 3550, IETF Network Working Group, July 2003.

[Sch96] B. Schneier. *Applied Cryptography.* John Wiley & Sons, New York, NY, USA, 2nd edition, 1996.

[SE02] D. Stanley and J. Edney. Minutes of IEEE P802.11i plenary session, Sydney, New South Wales, Australia, May 2002.

[Sim96] W. Simpson. PPP Challenge Handshake Authentication Protocol (CHAP). RFC 1994, IETF Network Working Group, August 1996.

[Stå00] M. Ståhlberg. Radio jamming attacks against two popular mobile networks. In H. Lipmaa and H. Pehu-Lehtonen, editors, *Proceedings of the Seminar on Network Security: Mobile Security.* Telecommunications Software and Multimedia Laboratory, Department of

Computer Science and Engineering, Helsinki University of Technology, December 2000.

[Sti02] D. R. Stinson. *Cryptography: Theory and Practice*. Chapman & Hall/CRC Press, Boca Raton, FL, USA, 2nd edition, February 2002.

[SW89] J. Simpson and E. Weiner, editors. *Oxford English Dictionary*, volume 14. Oxford University Press, Oxford, England, UK, 2nd edition, 1989.

[USPt02] Method and apparatus for facilitating efficient authenticated encryption. United States Patent Application 20020071552, June 2002.

[VMS04] J. P. Vila, N. Mathur, and L. Sanchez. Requirement Specification for MAC/RRM. IST Project 507102, Deliverable 3.3.1, Version 1.0, June 2004.

[WC81] M. N. Wegman and J. L. Carter. New hash functions and their use in authentication and set equality. *Journal of Computer and System Sciences*, 22(3):265–279, June 1981.

[WGB98] D. Wagner, I. Goldberg, and M. Briceno. GSM cloning, April 1998. [referenced on September 1, 2004]
`http://www.isaac.cs.berkeley.edu/isaac/gsm-faq.html`.

[WHF03] D. Whiting, R. Housley, and N. Ferguson. Counter with CBC-MAC (CCM). RFC 3610, IETF Network Working Group, September 2003.

[WW02] M. Walker and T. Wright. Security. In F. Hillebrand, editor, *GSM and UMTS: The Creation of Global Mobile Communication*, chapter 15. John Wiley & Sons, Chichester, England, UK, 2002.

Appendix A

Description of CCM Mode of Operation

This appendix presents the AEAD-type CCM mode of operation used by WLAN, IEEE 802.15.3, and IEEE 802.15.4. Currently, the mode is specified only for 128-bit block ciphers, such as AES. There are also two parameters that can be chosen to suit the needs of the application:

- length of the message authentication value in octets, which is denoted by $M \in \{4, 6, 8, 10, 12, 14, 16\}$, and

- maximum length of the message characterized by $L \in \{2, 3, 4, 5, 6, 7, 8\}$, the maximum length being $2^{8L} - 1$ octets

The byte order in CCM integer encoding is *most significant byte first*.

A.1 Definitions

Table A.1 shows the notation used in the following discussion. The notation used here is compatible with that of the CCM specification document [WHF03]. However, the following presentation is a bit more formal than that.

Symbol	Meaning
ε	String of length 0
$l(s)$	Length of string s in octets
s_i	ith 16-octet (128-bit) block of string s (where $1 \leq i \leq \frac{l(s)}{16}$)
$s \oplus t$	Bitwise xor of two strings (s and t) of equal length
$s\|\|t$	Concatenation of strings s and t
$str_n(i)$	n-bit string representation of non-negative integer i (MSB first)
$trunc_n(s)$	n first octets of string s
$E(K, s)$	Encryption of 128-bit string s using key K

Table A.1: Notation used in CCM description

Note that s_0 is not considered to be part of string s but a distinct constant. This makes the equations reasonably convenient while still preserving compatibility with the CCM specifications.

In addition, we define the following constants and functions:

$$\begin{aligned}
M' &= \frac{M-2}{2} \\
L' &= L-1 \\
z^n &= str_n(0) \\
pad(s) &= \begin{cases} s & \text{if } l(s) \bmod 16 = 0 \\ s||z^{128-8\cdot(l(s) \bmod 16)} & \text{otherwise} \end{cases}
\end{aligned} \tag{A.1}$$

A.2 Inputs

CCM mode requires the following inputs:

1. Encryption key K, the format of which is determined by the block cipher.

2. Unique bitstring (nonce) N that has not been used with the same value of K, such that $l(N) = 15 - L$.

3. Plaintext message m satisfying condition $0 \leq l(m) < 2^{8L}$.

4. AAD affecting the authentication value. AAD is denoted by a and satisfies condition $0 \leq l(a) < 2^{64}$.

A.3 Encryption and Authentication

Authentication field T is computed as shown in Figure A.1. More formally:

$$\begin{aligned}
a' &= \begin{cases} \varepsilon & \text{if } l(a) = 0 \\ str_{16}(l(a))||a & \text{if } 0 < l(a) < 2^{16} - 2^8 \\ str_{16}(2^{16}-2)||str_{32}(l(a))||a & \text{if } 2^{16} - 2^8 \leq l(a) < 2^{32} \\ str_{16}(2^{16}-1)||str_{64}(l(a))||a & \text{if } 2^{32} \leq l(a) < 2^{64} \end{cases} \\
b &= \begin{cases} 0 & \text{if } a = \varepsilon \\ 1 & \text{otherwise} \end{cases} \\
B_0 &= z^1||str_1(b)||str_3(M')||str_3(L')||N||str_L(l(m)) \\
B &= pad(a')||pad(m) \\
X_0 &= 0 \\
X_{i+1} &= E(K, X_i \oplus B_i) \quad \text{where } 0 \leq i \leq \frac{l(B)}{16} \\
T &= trunc_M(X_{\frac{l(B)}{16}+1})
\end{aligned} \tag{A.2}$$

Keystream is generated using the following equations:

$$\begin{aligned}
A_i &= z^5||str_3(L')||N||str_L(i) \\
S_i &= E(K, A_i)
\end{aligned} \tag{A.3}$$

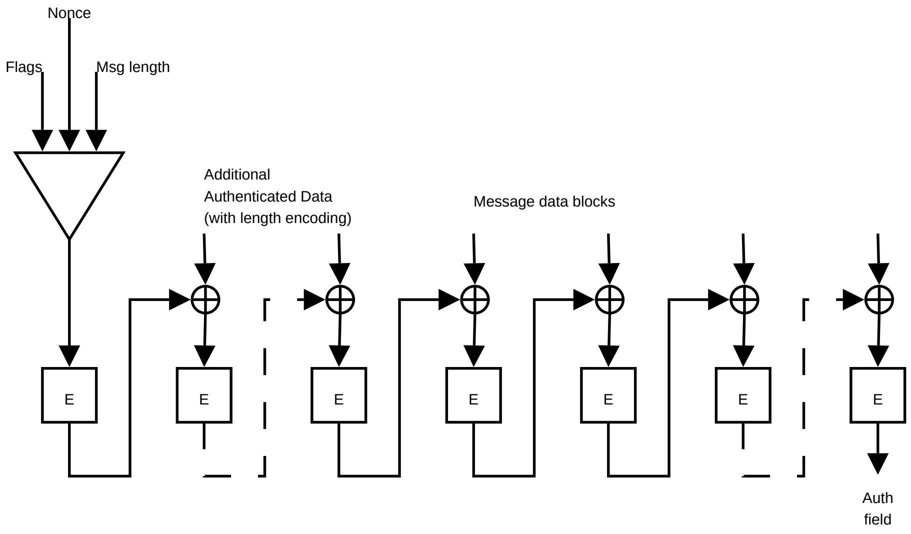

Figure A.1: CBC-MAC calculation

where $i \geq 0$. The output of CCM mode consists of ciphertext c and authentication value U, computed as follows:

$$\begin{aligned} c &= m \oplus trunc_{l(m)}(S) \\ U &= T \oplus trunc_M(S_0) \end{aligned} \tag{A.4}$$

The encryption process is illustrated in Figure A.2.

A.4 Decryption and Verification

Assume the recipient receives ciphertext c, AAD a, and possibly false authentication value U'. He also knows key K and nonce N. To decrypt c, he generates the keystream using Equation (A.3). Then he can decrypt c and U':

$$\begin{aligned} m &= c \oplus trunc_{l(c)}(S) \\ T' &= U' \oplus trunc_M(S_0) \end{aligned} \tag{A.5}$$

Knowing message m, the recipient can now compute the correct authentication field T by using Equation (A.2). If $T \neq T'$, the message is discarded.

A.5 Criticism

In addition to defining the maximum message length, parameter L determines the length of the nonce, such that the message length field and the nonce together occupy 15 octets in B_0. Naturally, L also determines the length of the counter field in A_i. What is strange is that the sizes of both length and block counter fields are equal to L octets. The maximum number of blocks is obviously less than the message length in octets, and therefore 15 counter values out of 16 cannot be used. For example, if $L = 2$, the maximum length of the

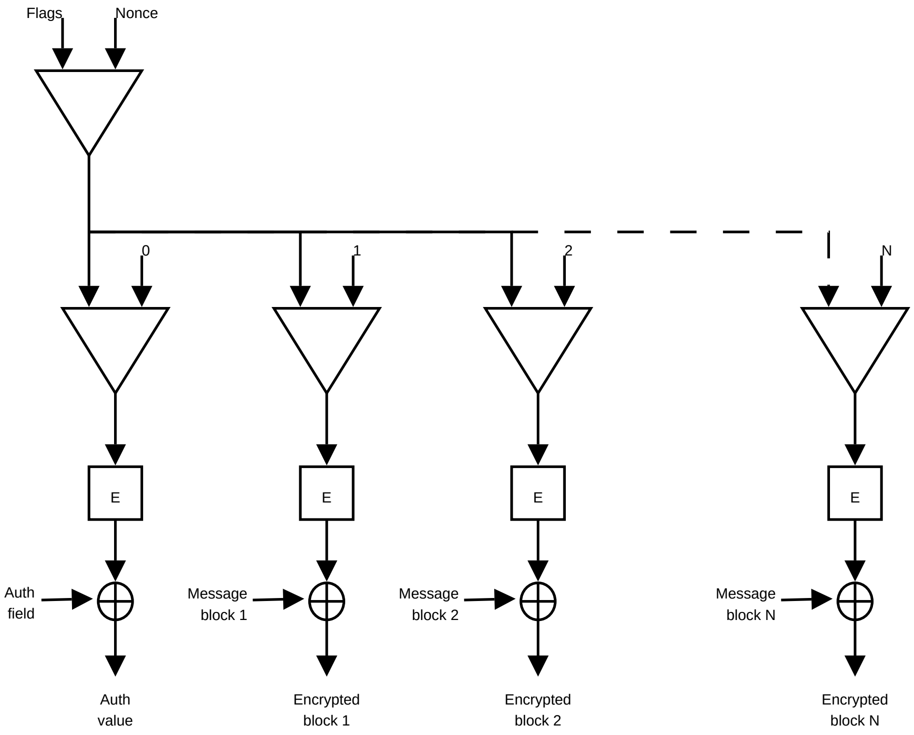

Figure A.2: Counter mode encryption

message is about 64 kB, although it could be 1 MB if there were $8L + 4$ bits in the length field instead of $8L$.

Of course, L can be increased to enable encryption of larger messages. But the disadvantage is that by increasing L, the length of the nonce is decreased. This peculiar connection between the nonce length and the maximum length of the message was criticized in [RW03, Sect. 3.2].

It would not have been necessary fit the block counter and the message length into the same space. For example, in GCM, an additional plaintext block containing the lengths of the message and AAD is appended to the message when computing the authentication value, thus not limiting the size of the nonce. In fact, GCM accepts nonces of arbitrary length by reducing them to the suitable size by applying a keyed hash function. [MV04b, Sect. 2]