SECURITY TECHNIQUES

Chris J Mitchell[1]

Introduction

In this paper we consider the range of security techniques available to future designers and implementors of telecommunications networks. The primary focus of this paper will be mobile networks, which is where much of the recent public effort on telecommunications security has been exerted. However, the techniques described are of general applicability in telecommunications.

Before proceeding we need to establish our terminology and, in particular, to define what we mean by a security technique; in doing so we will distinguish between security *features* and security *techniques* (or security *mechanisms*, as security techniques are often known). Security provisions in systems are present, not for their own sake, but to combat identified security *threats*. To combat these threats requires the provision of specific security *features* (sometimes known as *services*), such as

- *confidentiality* - to address the threat of unauthorised disclosure of information by means of eavesdropping, etc.,
- *data integrity* - to address the threat of unauthorised modification to information,
- *origin authentication* - to address the threat of information being spuriously inserted into a network,
- *entity authentication* - to address the threat of one entity masquerading as another,
- *non-repudiation* - to address the threat of an entity repudiating its actions (i.e. denying actions it has taken).

These features exist as abstract concepts, and are independent of the means used to provide them. Features are provided by security *techniques* (or *mechanisms*), which include

- *encipherment algorithms* - used to help provide confidentiality features,
- *integrity mechanisms* - (which include the well-known MACs), used to help provide data integrity and origin authentication features,
- *digital signature algorithms* -which can be used to help provide non-repudiation features,
- *authentication exchanges* - used to help provide entity authentication features.

At this point we should mention ISO 7498-2: 1989, which provides a standardised language for discussing security features and techniques (or security services and mechanisms as ISO 7498-2 describes them). We will use the terminology defined in ISO 7498-2 wherever possible, although we prefer to use the term security feature rather than security service to avoid confusion with services provided via a telecommunications network. ISO 7498-2 defines a range of types of security feature, as well as a classification of security techniques, and describes which techniques might be used to provide each of the defined security features.

In looking at techniques appropriate to telecommunications networks we will summarise the progress which has been made in recent years in providing general-purpose standards for security techniques. This work has primarily taken place within ISO/IEC JTC1/SC27/WG2.

Security services

In order to put our discussion of security techniques into an appropriate context, it is first helpful to consider what types of security feature are required in telecommunications networks. To do this we first review the security features provided in the current `second generation' mobile telecommunications standards for GSM and

[1] Information Security Group, Royal Holloway, University of London, Egham, Surrey TW20 0EX.

DECT (produced by ETSI). We then briefly review possible requirements for security features in future mobile telecommunications networks.

In the GSM standards, the following security features are provided:

- *confidentiality* of user and signalling data passed across the air interface,
- *confidentiality* for the user identity when passed across the air interface, and
- *entity authentication* between a mobile and a base station (across the air interface).

The DECT standards provide a similar set of security features, i.e. they are also restricted to air interface confidentiality and entity authentication.

Standards for future mobile telecommunications networks will almost certainly need to allow for the provision of a wider range of security features; in addition, the security techniques specified in the GSM and DECT standards may be inappropriate for future mobile networks. Examples of additional security features which might be supported in future mobile networks standards (ETSI UMTS and ITU-T FPLMTS) include:

- *integrity* for data passed across the air interface,
- end-to-end data *confidentiality* between mobiles,
- *non-repudiation* of charging information.

Of course, just because certain security features and supporting mechanisms are not specified in GSM, does not mean that they cannot be provided in GSM networks. However, because these features are not specified in the standards, proprietary solutions are needed, which can impose considerable extra costs on Network Operators in designing additional security facilities, and in commissioning their implementation in network equipment. This means there is a considerable advantage, both to operators and manufacturers, in providing wider-ranging security standardisation for future networks.

Security techniques

We now consider a range of types of security technique which are relevant to the provision of the types of security feature we have discussed above. In particular we consider the following categories of security technique (note that some techniques can be used to help as components in other, more complex, techniques):

- *encipherment* techniques (for providing *confidentiality* features),
- *integrity* techniques (for providing *data integrity* features and as a building block in authentication exchanges),
- *digital signature* techniques (for providing *data integrity* and *non-repudiation* features),
- *hash-functions* (used in conjunction with digital signatures and also as a means of building integrity techniques), and
- *authentication exchanges* (for providing *entity authentication* features).

GSM and DECT make use of encipherment techniques (to provide air interface data confidentiality and user identity confidentiality), integrity techniques (to help construct authentication exchanges), and authentication exchanges (to provide entity authentication).

We now look at each of these categories of mechanism in a little more detail.

Encipherment techniques

No international standards exist for encipherment techniques, despite the US standardisation of the *DES* block cipher algorithm some 15 years ago (ANSI X3.92-1981), and its subsequent adoption as a de facto standard by the international banking community. Instead work has focused on creating an international *register* of encipherment algorithms, which will provide each registered algorithm with a unique name. The form of register entries, and the procedure for having entries added to the register, is standardised in ISO/IEC 9979: 1991. The international *Registration Authority* is the NCC here in the UK

The GSM standards make use of a proprietary algorithm (called *A5*), which is owned by ETSI, and the details of which are kept secret, and released only to authorised manufacturers who are obliged to sign non-disclosure agreements. The A5 algorithm is not currently on the register (as of November 1994 there were ten registered algorithms), although there is no reason why it should not be, since registration of an algorithm does not require any disclosure of the algorithm's operation. Given the status of international standardisation for encipherment techniques, and the long-standing political sensitivity of encipherment technology, the GSM approach appears to be the only sensible way ahead, and for the foreseeable future this appears to be the most appropriate means for selection of encipherment techniques.

Before proceeding it is worth observing that, although there are no internal encipherment algorithm standards, there is a standard governing ways in which a block cipher algorithm (such as DES) might be used. This *modes of operation* standard (ISO/IEC 10116: 1991), has evolved from the ANSI standard prescribing modes of use for the DES block cipher (ANSI X3.106-1983). However, this ISO/IEC standard is probably of limited relevance to telecommunications networks, since high-speed stream ciphers rather than block ciphers are probably appropriate for the majority of telecommunications networks (it is certainly true that A5 is a stream cipher algorithm).

Integrity techniques

The purpose of a data integrity technique is to enable the recipient of a data string (protected using the technique) to both verify its origin and that it has not been changed in transit. Hence such a mechanism can be used to provide both data integrity and origin authentication features.

Standards for integrity mechanisms date back to the early 1980s; ANSI has published two integrity mechanisms for banking use (X9.9-1986 (revised) and X9.19-1985). A corresponding international banking standard, ISO 8731-1: 1987, has subsequently been published. All three of these standards specify use of the DES block cipher algorithm in Cipher Block Chaining (CBC) Mode to produce a *Message Authentication Code (MAC)*. A different integrity mechanism, called the *Message Authenticator Algorithm (MAA)*, is specified in the banking standard ISO 8731-2: 1992 (second edition). Following from the banking work, the international standard ISO/IEC 9797: 1994 (second edition) for an integrity mechanism has been produced; this technique is also based on the use of a block cipher in CBC mode.

The GSM and DECT standards do not support the provision of origin authentication or data integrity services. However they both make use of an integrity mechanism as part of an authentication exchange technique which supports entity authentication of a mobile to a base station (we discuss this further below). Future telecommunications networks may well wish to offer a data integrity service as an optional feature of service provision; it is also highly possible that ISO/IEC 9797 will not be appropriate for use, as it requires the implementation of a block cipher. However, one-way hash-function algorithms are at an advanced stage of standardisation (see below), and they can be simply adapted for use as integrity mechanisms, and this might well provide a possible route for future telecommunications networks. It certainly seems reasonable to suppose that future network designers will wish, wherever possible, to user standardised cryptographic techniques rather than design their own.

Finally it is important to note that, in parallel with the cryptographic *check functions* of the type described, the general class of integrity mechanisms also includes non-cryptographic mechanisms which are vital to the provision of integrity protection for sequences of data packets, for each of which an individual MAC or check-value may be computed. Examples of these non-cryptographic mechanisms include the use of sequence numbers or timestamps to provide data integrity for entire sequences of packets against manipulation (including threats such as duplication and deletion of entire packets).

Digital signature techniques

A digital signature technique is a function which, when applied to a data string, produces a result which enables the recipient to verify the origin and integrity of a data string, i.e. it can be used to provide data integrity and origin authentication features. Moreover it has the property that only the originator of a data string can produce a valid signature, i.e. being able to verify the correctness of a signature produced by an entity, does not provide the means to compute that entity's signature on any data string. Digital signature techniques can be used to provide *non-repudiation* of origin for a message, i.e. the recipient of a message with entity *A*'s signature on it, has evidence that *A* sent the message which even *A* cannot repudiate.

Digital signature algorithms can be divided into two types.

- Digital signatures *with message recovery* - which have the property that the message can be recovered from the signature itself; such signatures can normally only be applied to messages of limited length (e.g. of length at most 500 bits).
- Digital signatures *with appendix* - where the message needs to be sent in parallel with the signature, and signature acts as a `check' on the separately transmitted message.

A technique of the first type has been standardised in ISO/IEC 9796: 1991. The algorithm used is a variant of the well-known *RSA* algorithm, the first and best known of the *public key* (or *asymmetric*) cryptographic algorithms.

A US standard was recently published (the NIST *Digital Signature Standard (DSS)*) for a different signature algorithm (a variant of the El Gamal signature algorithm), this time of the second, and more generally useful, type. This technique is almost certain to be included in an emerging ISO/IEC standard, which will contain several techniques for Digital signatures with appendix. In parallel with these `general purpose' standards, US and international banking standards committees have been developing standards incorporating the use of RSA signatures.

It is therefore now possible to say that there are a variety of well-established signature techniques available. However, they all share the same implementation difficulties (albeit to varying extents), namely that calculation of a digital signature can be very computationally intensive. This means that, until very recently, their implementation in hand-portable telecommunications terminals has been impractical. Smart cards capable of performing digital signatures in a small fraction of a a second, which is what is needed, have proved difficult to make at an economic price.

However, as soon as such devices can be made at or close to the price of current smart cards, the use of digital signature techniques could become very attractive. As well as providing all the features that a conventional integrity mechanism, such as those built into GSM and DECT, can offer, key management for digital signature techniques is significantly simpler, since verification keys need not be kept secret. Moreover, digital signature techniques enable the provision of non-repudiation features, which in turn makes telecommunications services such as *irrefutable charging* and/or non-repudiable financial transactions a possibility.

One-way hash-functions

One-way hash-functions are an essential component of all digital signature with appendix techniques. A hash-function takes as input a data string of arbitrary length, and outputs a *hash-code* of some small fixed length (e.g. 128 bits). In the context of a digital signature with appendix technique, the message to be signed is first input to a hash-function, and then the derived hash-code is input to the signature algorithm itself. However, hash-functions also have uses outside the context of digital signature algorithms. One important characteristic of hash-functions is that, unlike most other cryptographic functions, they do not use a secret key, and hence their entire operation is typically public.

A multi-part international standard for one-way hash-functions (ISO/IEC 10118) is under development. The first two parts (ISO/IEC 10118-1: 1994 and ISO/IEC 10118-2: 1994) have now been published. Part 1 contains general information, and Part 2 describes two methods for generating a hash-function from a block cipher.

Part 3, for which publication is planned in 1996, specifies in complete detail two ‘dedicated’ hash-functions, both designed specifically for the purpose. One, known as *SHS*, is already the subject of a recently published US NIST standard; it is designed to be used with the NIST DSS (signature) algorithm, although it is equally applicable to other signature algorithms. The other, known as *RIPEMD*, has been developed as part of the EU-funded RIPE project; RIPEMD is itself an ‘improved’ version of MD4, a hash-function developed by RSA Inc. Both of these algorithms are simple to implement in software, and can process data at high rates even on modest microprocessors.

Part 4, also likely to be published in 1996, specifies a pair of hash-functions based on modular exponentiation. The reason for designing such hash-functions is that some digital signature algorithms (e.g. RSA) are also based on modular exponentiation, and hence a hash-function which could make use of the same arithmetic functions, perhaps implemented in hardware, might be very advantageous.

Parts 3 and 4 are most likely to be of value in telecommunications networks, in particular either as part of a digital signature algorithm, or as the basis of an integrity mechanism. In fact it is very simple to use a hash-function to produce an integrity mechanism; the data to be integrity protected is concatenated with a secret key, and the resulting data string is input to the hash-function. The hash-code output can then serve as a check-value on the original data string, and, given the hash-function is itself cryptographically strong, the check-value for a message can only be calculated by someone possessing the correct secret key.

Finally we note one other possible application for a one-way hash-function which may be of relevance to telecommunications networks. If a stored data object (e.g. a file) needs protection against change, it can be input to a hash-function and the output stored in a secure place. Recomputing the hash-code and comparing it with the stored value can be used to verify the correctness of the stored data. This is valuable in protecting against malicious changes made by malicious entities or programs (e.g. viruses).

Authentication exchanges

Authentication exchange techniques (or *authentication protocols* as they often called) are exchanges of cryptographically protected messages, which have the objective of enabling two communicating entities to verify one another’s identity, i.e. they provide *entity authentication* features. Entity authentication can only be achieved for a single instant of time. Typically these exchanges are used to initiate a secure connection, and *Session Keys* may be established as a by-product of the authentication process.

In order for these authentication exchanges to work, some or all, of the messages need to be protected by a cryptographic mechanism, typically either an encipherment mechanism, a digital signature or an integrity mechanism. In addition, non-cryptographic mechanisms (typically time-stamps or random ‘nonces’) need to be included in the messages to guarantee that the messages are ‘fresh’.

The first standardised authentication exchanges were specified in CCITT X.509 (1988) (ISO 9594-8), which contains three different protocols all based on digital signature techniques; a revised version of this ITU-T Recommendation/ISO standard has recently been published. Since then a multi-part standard ISO/IEC 9798 has been developed, containing a variety of different authentication exchange techniques using a range of different underlying cryptographic mechanisms.

Part 1 (ISO/IEC 9798-1: 1991) contains a general model for techniques of this type. Part 2 (ISO/IEC 9798-2: 1994) contains a set of authentication exchanges in which the messages are protected using an encipherment technique. Part 3 (ISO/IEC 9798-3: 1993) contains another set of authentication exchange techniques. These techniques are all based on the use of digital signature techniques to cryptographically protect the messages. Yet another set of authentication exchange techniques, this time based on the use of integrity mechanisms, is to be found in Part 4 (ISO/IEC 9798-4: 1995). In each of Parts 2, 3 and 4, some of the exchanges are suitable for use when synchronised clocks are available (and hence the messages contain time-stamps which can be used to verify their ‘freshness’), and some use random or pseudo-random *nonces* (or *challenges*) instead.

The GSM and DECT standards both specify the use of authentication exchange mechanisms to provide entity authentication services between a mobile and a base station (over the air interface). In both cases, although the mechanisms are slightly different, the mechanisms conform to techniques specified in ISO/IEC 9798-4, i.e. the mechanisms are based on the use of integrity techniques. In both cases freshness is guaranteed by the use of pseudo-random nonces (challenges). The integrity techniques used are in both cases based on secret proprietary algorithms.

In both GSM and DECT the authentication exchanges are used at the start of a connection; they also provide the basis of session key distribution systems. The session key is subsequently used to encipher the data sent over the connection.

The use of authentication exchanges is fundamental to the provision of security in any situation where a communications link is liable to attack. It is difficult to imagine any future mobile networks not employing such a technique as a first line of defence against users wishing to fraudulently obtain service. Authentication exchanges may also be employed where different network operators an/or service providers need to verify one another’s identity when they are collaborating to provide service to a user.

Finally it is also important to observe that authentication exchange techniques can be integrated with techniques to provide user identity confidentiality based on the use of continually updated `temporary identities’. Such schemes already operate in GSM, and an elaborated version of the GSM and DECT schemes has recently been proposed for adoption as a mechanism for use in FPLMTS.

Summary

We conclude this paper by observing that it should be clear that there already exist an increasingly wide range of standardised security techniques of potential use in future telecommunications networks. In the future the existence of these standardised techniques may remove the need for the design of new techniques whenever a new system is created. In the medium term, and perhaps sooner than many experts would like, this may well obviate the need for specialised cryptographers, at least as far as commercial communications are concerned! The real need, both now and in the future, is for skilled staff capable of selecting, integrating and managing the ever-increasing range of security products and systems available.

Acknowledgements

In producing this paper, the author has benefited a great deal from work performed as part of the *Third Generation Mobile Telecommunications Systems Security Studies (3GS3)* Project, funded by EPSRC Grant GR/J17173 under the auspices of the DTI/EPSRC LINK Personal Communications Programme. The author would like to acknowledge the major part played in this project by the industrial partners: GPT Ltd. and Vodafone Ltd. It should nevertheless be stressed that the views in this paper are the author’s own, and do not necessarily represent the views of GPT, Vodafone or any other body.