Is entity authentication necessary?

Chris J. Mitchell and Paulo S. Pagliusi

Information Security Group, Royal Holloway, University of London
Egham, Surrey TW20 0EX, UK
c.mitchell@rhul.ac.uk   p.s.pagliusi@rhul.ac.uk

Abstract. Conventionally, mutual entity authentication is seen as the necessary precursor to the establishment of a secure connection. However, there exist examples of cases where authentication is not needed. The purpose of this paper is to consider this proposition, illustrated by case studies, and to use the findings of this investigation as input for the design of authentication protocols suitable for use in future Internet access environments supporting ubiquitous mobility.

1 Introduction

In the context of secure communications, mutual entity authentication is very commonly seen as the necessary precursor to the establishment of a secure connection. However, there do exist examples of cases where mutual authentication is not necessary, and, indeed, may impose unnecessary overheads on session establishment. The purpose of this paper is to consider this proposition, using case studies as the basis for this discussion. In these case studies we consider the protocols used in the GSM (Global System for Mobile Communications[1]) and 3GPP (3rd Generation Partnership Project[2]) mobile telecommunications systems.

The main application context of these discussions covers the case where there are three entities involved in the authentication exchange: a mobile user, a local AAA (Authentication, Authorisation and Accounting) server, and a remote (home) AAA server. That is, the mobile user wishes to set up some kind of secure link with a 'local' network (with its own AAA server), and the mobile user has a long term cryptographic relationship, typically backed up by some kind of contractual and payment arrangement, with a remote (home) network and AA server. This 'roaming user' model is not only becoming an increasingly common model for Internet access, but it is also fundamental to understanding the air interface security system for present day mobile telecommunications networks (e.g. GSM and 3GPP).

The purpose of this paper is not so much to talk about GSM and 3GPP, but to consider what more general lessons can be drawn regarding future protocol design. In particular, how best should authentication and/or access security be

[1] http://www.gsmworld.com

[2] http://www.3gpp.org

designed in the scenario where a mobile user wishes to access the Internet via a multiplicity of different network types? The recently inaugurated IETF PANA (Protocol for carrying Authentication for Network Access[3]) work will provide a general framework for the exchange of authentication messages in this mobile scenario, but will not address the question of exactly how access security should operate. Other relevant work includes the ongoing IST-Shaman project[4].

2 Entity authentication and key establishment

Before proceeding we need to establish some terminology. We use definitions from the Handbook of Applied Cryptography (HAC), [1], and, where relevant, we indicate the relevant section number from the HAC in brackets after the definition.

Entity authentication is the process whereby one party is assured of the identity of a second party involved in a protocol, and that the second has actually participated (10.1.1). Either one or both parties may corroborate their identities to each other, providing, respectively, *unilateral* or *mutual* authentication (10.1.2).

We are particularly concerned here with the case where a protocol simultaneously provides entity authentication (unilateral or mutual) and session key establishment, where this session key (or keys) is used to protect data subsequently transferred. *Key establishment* is a process or protocol whereby a shared secret becomes available to two or more parties, for subsequent cryptographic use (12.2.1). *Key authentication* (sometimes also called *implicit key authentication*) is the property whereby one party is assured that no other party aside from a specifically identified second party (and possibly additional identified trusted parties) may gain access to a particular secret key (12.2.1). *Key confirmation* is the property whereby one party is assured that a second party actually has possession of a particular secret key (12.2.1). *Explicit key authentication* is the property obtained when both (implicit) key authentication and key confirmation hold (12.2.1).

A further property, desirable in some practical applications but not discussed in [1], is *key freshness*. By this we mean the property that the party to a key establishment process knows that the key is a 'new' key. In particular, the party should have evidence that the messages received during the protocol by which the key has been established are 'fresh' messages, i.e. they are not replays of 'old' messages from a previous instance of the protocol.

To see why this property is necessary in addition to implicit or explicit key authentication, consider the following very simple one-pass key establishment protocol. In this protocol we suppose that A and B share a long term secret key K. Entity A chooses a session key K_s and sends it to B in the following message:

$$e_K(K_s||I_B)$$

[3] `http://www.ietf.org/html.charters/pana-charter.html`

[4] `http://www.ist-shaman.org`

where $e_K(X)$ denotes the encryption of data string X using key K, I_B is an identifier for party B and —— denotes concatenation of data items. Note that we suppose here that the encryption algorithm also provides message integrity and origin authentication (e.g. by additionally computing a Message Authentication Code (MAC) using a variant of K — see, for example, [1]).

This protocol clearly provides (implicit) key authentication to B, given that we assume that K is known only to A and B. It also provides key confirmation to B, since the inclusion of I_B means that the message originates from A, and hence A must know K_s. However it should be clear that the protocol does not provide key freshness, since B has no way of telling whether the message has just been generated by A, or is a replay of a message sent by A at any time since K was first established. Of course, this lack of key freshness can easily be rectified by including a time stamp or sequence number within the scope of the encrypted message.

The absence of key freshness would enable an interceptor to force B to keep re-using an 'old' session key, which might have been compromised. It would therefore seem reasonable to make key freshness a requirement for most applications of key establishment protocols. In fact it turns out that the absence of key freshness is a possible source of weakness in the GSM protocol, as we discuss below.

To conclude this discussion, we note that the two critically important properties for most key establishment protocols would appear to be (implicit) key authentication and key freshness. *Explicit* key authentication is not always so important, and is, in any case, achieved once a party receives evidence of use of a key.

3 Case study I: GSM

We start by considering the GSM air interface security features. For a more detailed discussion of these features see, for example, [2] or [3]. Note that we are concerned here exclusively with the protocol design for GSM, and not with the security of the algorithms used; for a summary of the current security status of the GSM algorithms see, for example, [2].

3.1 Outline of scheme

The GSM air interface authentication protocol, i.e. the security protocol used across the wireless path between mobile and network, takes place between a mobile telephone (actually the Subscriber Identity Module (SIM) within the telephone) and the network with which it is currently registered, i.e. the network which the mobile is using to make and receive calls. This is performed with the assistance of the mobile user's 'home network', which originally supplied the SIM. These three entities respectively fit the roles of mobile user, local AAA server, and home AAA server described above.

The SIM and the home network share a long term secret key, K_i. When a mobile user first registers with a new network, this network approaches the home network of the user to request the information necessary to perform the air interface authentication protocol. This information is provided in the form of 'triplets', (N, R, K_c), where N is a random challenge (or nonce), R is the expected response to this challenge, and K_c is a secret session key, to be used to encrypt voice data exchanged between the mobile and the visited network[5]. Both R and K_c are computed as functions of N and the long-term secret key K_i.

To conduct the air interface authentication protocol, the visited network sends the mobile the challenge N. The mobile uses its stored value of K_i to compute R and K_c, and sends R back to the network. The network compares the received value of R with the value in the triple provided by the home network, and if the two agree then the mobile is deemed to have been authenticated. If encryption of the air interface link is required (this decision is made by the network) then the key K_c is used for this purpose.

3.2 Properties of scheme

We provide only a brief analysis of the scheme. For a more detailed analysis see, for example, [4] or [2].

First observe that the long term secret key K_i is not passed to the visited network. This to some degree limits the trust required in the visited network. Also, since the computation of R and K_c from N and K_i is only performed by the SIM and the home network, the cryptographic algorithms used can be network specific.

The protocol provides unilateral authentication of the mobile to the local network. The protocol also provides (implicit) key authentication to the mobile user, since the key K_c is computed using a long term secret known only to the SIM and the home network. However, the protocol does not provide key freshness to the mobile, since the mobile has no way of determining whether N is a 'fresh' challenge. Indeed, when the local network has run out of triplets and cannot, for some reason get access to any more triplets from the home network, it is allowed to re-use them.

3.3 Analysis

It has been stated on many occasions that the fact that GSM only provides unilateral authentication, i.e. of mobile to base station but not vice versa, is to blame for certain known security weaknesses with GSM. These weaknesses have been widely documented — see, for example, [1, 4]. They include the possibility of a false base station interposing itself between the genuine base station and the mobile, and using this to monitor all traffic passing to and from the mobile.

[5] N and R are commonly referred to as 'RAND' and 'XRES'/'SRES' respectively

However, detailed analysis of these GSM weaknesses reveals that adding base station authentication to GSM will not necessarily prevent these problems. A false base station could still insert itself between the mobile and genuine base station *after* a mutual authentication process, since there is no integrity protection for the exchanged traffic. Note also that adding routine integrity protection to the air interface link would not really be viable, since the air interface link is subject to a high level of errors. Deleting all traffic containing errors is not an acceptable strategy, since digitised speech can 'survive' a modest number of transmission errors; that is, enforcing integrity protection would potentially transform a poor quality but usable speech channel into no channel at all.

In fact, the interposition attack would normally be made pointless by routine GSM encryption. However, problems arise because the base station instructs the mobile whether or not to use encryption. A 'false' base station can therefore tell the mobile not to employ encryption, which gives rise to one of the main causes of weakness in the GSM scheme.

One solution to this problem is to provide integrity protection for certain security-critical signalling messages sent across the air interface, notably including the 'cipher enable/disable' messages, and this is precisely the solution adopted in 3GPP (see below). This integrity protection can be based on the session key (or keys) established during the air interface authentication protocol. This leads us to a second problem with GSM, namely the lack of key freshness for this protocol already mentioned above.

This lack of key freshness means that if a malicious third party ever obtains a valid triplet (N, R, K_c) for a mobile, then this can be used to launch an effective false base station attack without suppressing encryption on the radio path. This is discussed in more detail in [4].

4 Case study II: 3GPP

We next consider a much more recent air interface authentication protocol, designed as part of the 3GPP system. This protocol, specified in [5], has very similar objectives to the GSM protocol and, like GSM, is based on secret key cryptography. In fact, the design of the 3GPP protocol is closely based on the GSM scheme, taking into account the known problems with the GSM protocol.

4.1 An enhanced GSM protocol

Before giving a description of how 3GPP operates, we give an example of an 'enhanced' version of GSM. This example is not a serious proposal for adoption, but is intended to help explain the details of the 3GPP design, and what could go wrong if some of the features were not present.

Suppose that the mobile user's 'User Services Identity Module' (USIM), i.e. the successor to the GSM SIM, is equipped with a sequence number S (initially set to zero) as well as a long term shared secret K_i. Instead of providing triplets to the visited network, the home network provides 'quadruplets' (N, R, K_c, S), where N is as before, R and K_c are as before except they are a function of S (as well as N and K_i), and S is a sequence number. The home AAA server keeps a record of S for each mobile user, and generates quadruplets with monotonically increasing sequence numbers. (Note that the same effect can be achieved without keeping a database of sequence numbers - see, for example, [6]).

To authenticate the mobile, the visited network sends the challenge and serial number to the mobile (i.e. N and S). As long as S is larger than any previously received sequence number, the mobile accepts it, updates its stored sequence number, and computes the response R (and the session key K_c) as a function of N, S and K_i. This revised protocol now provides key freshness, since the mobile can check that S is fresh and moreover the session key is a function of S.

To address the other problem with GSM discussed above, a second session key could also be derived at the same time, and used to protect the integrity of security-critical signalling messages. By this means the major weaknesses of GSM could be addressed.

However, there is one major problem with this solution. That is, there is a trivial and fatally serious denial of service (DoS) attack. An attacker can simply send a very large (maximal) sequence number S to a mobile, along with an arbitrary challenge N. The mobile will set its stored sequence number to this very large value, and will thereafter never accept any more challenges from the genuine base station. The existence of this DoS attack motivates the slightly more complex design of the actual 3GPP protocol, which we now describe.

4.2 Outline of 3GPP authentication

As in the 'enhanced GSM' protocol, the mobile user's USIM is equipped with a sequence number S (initially set to zero) as well as a long term shared secret K_i. Instead of providing triplets to the visited network, the home network provides '6-tuples' (N, R, K_c, K_a, S, M), where N, R and K_c are as in GSM, K_a is a second session key derived like K_c as a function of N and K_i (the long term secret key), S is a sequence number, and M is a MAC computed with data input a concatenation of N and S, and with secret key input K_i, the long term secret key[6]. As above, the home AAA server keeps a record of S for each mobile user, and generates 6-tuples with monotonically increasing sequence numbers.

To authenticate the mobile, the visited network sends the challenge, serial number and MAC to the mobile (i.e. N, S and M). The mobile first checks the MAC. If the MAC verifies correctly, and as long as S is larger than any previously received sequence number, the mobile accepts it, updates its stored sequence number, and computes the response R (and the session keys K_c and K_a) as a function of N and K_i.

[6] N, R, K_c, K_a, S and M are commonly referred to as 'RAND', 'XRES'/'SRES', 'CK', 'IK', 'SQN' and 'MAC' respectively, and the concatenation of S and M is commonly referred to as 'AUTN'. Note also that the above is a slightly simplified description of 3GPP authentication — in the actual scheme the sequence number S is sent encrypted to prevent it revealing the identity of the mobile user

The 3GPP protocol provides key freshness, since the mobile can check that S is fresh and moreover the challenge N is 'bound' to S by the MAC M. Thus, since the session keys are functions of N, they are also guaranteed to be fresh.

The session key K_c is used to encrypt data sent across the channel (just as in GSM) and K_a is used in parallel to protect the integrity of security-critical signalling messages. By this means the major weaknesses of GSM are addressed.

4.3 Properties of the 3GPP protocol

The 3GPP protocol clearly does not suffer from the DoS attack to which the 'enhanced GSM' protocol is prone, because of the presence of the MAC M. The 'false base station in the middle' problems are removed by providing data integrity and data protection for security-critical signalling messages from the base station to the mobile, including the message to enable encryption (this latter message is mandatory, so simply deleting it will not be an effective attack).

It is also worth noting that providing integrity protection for signalling messages without simultaneously providing key freshness would not be sufficient to deal with attacks arising from compromise of old session keys.

Finally note that it is the provision of key freshness combined with signalling integrity, not local network authentication, that prevents the false base station attacks to which GSM is prone. Nevertheless, and unlike GSM, mutual authentication is provided in 3GPP. This is because the inclusion of a MAC in the message sent to the mobile from the network enables the mobile to authenticate the source of the message, and the sequence number enables the message to be checked for freshness.

This would appear to undermine the main thesis of this paper, i.e. that mutual entity authentication is not always necessary. However we claim that the provision of network authentication to the mobile in 3GPP is essentially an accident. That is, it is provided only as an accidental by-product of the provision of other necessary security services. To support, this claim, first observe that the 'enhanced GSM' protocol in Section 4.1 does potentially deal with the main GSM problems, as long as it is used in conjunction with signalling integrity - the only problem with this protocol is the 'new' issue of a serious DoS attack. Secondly, in the next section we present a protocol which meets all the requirements of the 3GPP scenario without providing authentication of the network to the mobile and which is not prone to a DoS attack.

4.4 Other remarks

In fact there are additional advantages of the 3GPP protocol structure (as compared to the 'enhanced GSM' protocol), specific to the 3GPP operational environment. One potential advantage of sending the MAC M (apart from DoS prevention) is that it avoids the need to use S in calculating the session keys. This might be important for GSM/3G interoperation. In the 3GPP protocol, where the session keys are based on N and the long term secret K_i only, the GSM authentication triplet can be derived from the 3G authentication 6-tuple by simply ignoring S and M, and using a simple conversion function to derive a 64-bit GSM K_c from the two 128-bit keys K_c and K_a. This is important because dual mode mobiles will have to work in GSM networks which cannot handle 3G authentication 6-tuples.

One other reason for the MAC M is that it can be used to protect the 'AMF' field. AMF is an undefined data string concatenated with S which may be used for operator specific commands to the card.

5 Case study III: 3GPP-like protocols

We now briefly consider another 3GPP-like protocol. The main motivation for presenting this protocol is to show that the distinction between the need for specific properties for an established key and mutual authentication is real. In particular, because 3GPP provides mutual authentication, the suspicion might arise that the only way in which key freshness and key authentication can sensibly be obtained is to use a mutual entity authentication protocol.

The mobile user's USIM is here assumed to possess a clock synchronised to the clock of the network, as well as a long term shared secret K_i. Instead of providing 6-tuples to the visited network, the home network provides '5-tuples' (N, R, K_c, K_a, T), where N and R are as in 3GPP, K_a and K_c are derived as a function of N, T and K_i (the long term secret key), and T is a timestamp. This scheme requires 5-tuples to be used within a short period of their generation, as the mobile will check that T is current.

To authenticate the mobile, the visited network sends the challenge and timestamp to the mobile (i.e. N and T). The mobile first checks the timestamp to see if is within the 'window of acceptance'. If so, the mobile accepts it and computes the response R (and the session keys K_c and K_a) as a function of N, T and K_i.

The 3GPP protocol provides key freshness, since the mobile can check that T is fresh and moreover the session keys are functions of T. Just as in 3GPP, the session key K_c is used to encrypt data sent across the channel (just as in GSM) and K_a is used in parallel to protect the integrity of security-critical signalling messages. Finally, this scheme is not prone to the DoS attack, since the mobile will not reset its clock.

However, the protocol clearly does not enable the mobile to authenticate the network. Thus clearly entity authentication is not always required in order to establish key freshness and key authentication. However, this protocol is not a serious candidate for use in the 3GPP scenario, since assuming synchronised clocks (and the accompanying management overhead) is not reasonable in this environment.

6 Future systems

One issue which we have not examined in detail here is the fact that, in any mobile user scenario, there would appear to be a need to delegate some access security functions from the 'home' AAA server to the visited network AAA functionality. The question remains open as to how best this should be done, especially bearing in mind possible anonymity requirements and trust issues. (Anonymity is an issue partially dealt with by GSM, and more thoroughly provided by 3GPP, but is beyond the scope of this paper).

Exactly what security services are really required for which protocols, in particular is mutual authentication a genuine requirement, and how much of the provision of these services should be delegated to the visited network? Whilst 3GPP might provide a model for a solution based on symmetric cryptography, how should we solve the same types of problem using asymmetric cryptographic techniques? This latter question appears to be of importance, because of the advantages to be gained from the use of public key techniques in scenarios where the number of entities proliferates, and there is no single model for establishing bilateral trust relationships, e.g. as in the PANA workgroup scenarios.

7 Concluding remarks

By considering a number of example protocols, we have provided evidence for the view that entity authentication is not always an essential precursor for the establishment of secure communications. In the case of GSM, it is often claimed that the lack of mutual entity authentication is the source of certain well known problems. However, it is clear that not only is this not the case, but also mutual authentication on its own will not solve the problems.

We have further argued that, in the typical case, the most important issue is to ensure that the properties of (implicit) key authentication and key freshness are provided for any established session keys. These session keys can be used to protect the integrity of security-sensitive data exchanged during the session, thereby preventing 'man in the middle' attacks.

Of course, key freshness can be obtained 'for free' if the key establishment process is embedded within a mutual entity authentication protocol. Indeed, it is this fact that may perhaps be responsible for the fact that the issue of key freshness is not widely discussed in the literature examining key establishment protocols (notably it is omitted from the discussion in [1]). However, the importance of the key freshness property means that, if protocols are designed which do not provide mutual authentication, then it is vital that the provision of key freshness is carefully checked.

8 Acknowledgements

The authors would like to acknowledge the many helpful insights and corrections provided by Peter Howard, Keith Martin, Kenny Paterson and Scarlet Schwiderski-Grosche. The authors would also like to thank all the participants in the IST Shaman project for their advice and encouragement.

References

1. Menezes, A., van Oorschot, P., Vanstone, S.: Handbook of Applied Cryptography. CRC Press, Boca Raton (1997)
2. Walker, M., Wright, T.: Security. In Hillebrand, F., ed.: GSM and UMTS: The creation of global mobile communication. John Wiley & Sons (2002) 385–406
3. Sutton, R.: Secure communications: Applications and management. John Wiley & Sons (2002)
4. Mitchell, C.: The security of the GSM air interface protocol. Technical Report RHUL-MA-2001-3, Mathematics Department, Royal Holloway, University of London, Egham, Surrey TW20 0EX, UK (2001) Available at `http://www.ma.rhul.ac.uk/techreports`.
5. Third Generation Partnership Project: 3GPP TS 33.102: Security Architecture. V3.11.0 edn. (2002) Technical Specification Group Services and System Aspects, 3G Security, Valbonne, France.
6. Mitchell, C.: Making serial number based authentication robust against loss of state. ACM Operating Systems Review **34** (2000) 56–59