Making serial number based authentication robust against loss of state

Chris J. Mitchell*
Information Security Group,
Royal Holloway, University of London,
Egham, Surrey TW20 0EX, U.K.

24th November 1999

Abstract

In this paper a novel method for making serial number based authentication protocols resilient against database failure is described. The method is applicable to the situation where a single server wishes to authenticate itself to a number of clients, but cannot reliably maintain a sequence number database. The solution has recently been included in a draft international standard for third generation mobile telecommunications, [1].

1 Introduction

Entity authentication, i.e. the verification by one entity of another entity's identity at a point in time, is a fundamental part of connection establishment in a variety of network scenarios. Authentication is often combined with key establishment, where the established key is used to protect data transferred during the connection.

Entity authentication protocols, and combined entity authentication and key establishment protocols, are based on two types of mechanism: cryptographic mechanisms and timeliness mechanisms. Cryptographic mechanisms are used to protect individual messages, and to enable the recipient of a message to verify its origin and integrity. Timeliness mechanisms enable the recipient of a message to verify its 'freshness'. This paper is concerned with a particular type of freshness mechanism, namely the use of 'sequence numbers' (or 'virtual timestamps', in the sense of Lamport, [3]).

*The research reported in this paper was carried out with the financial support of Vodafone Ltd. and the EU-funded ACTS project AC336 USECA (`http://www.useca.freeserve.co.uk`).

2 The state problem

As described in ISO/IEC 9798-1, [2], there are three main types of 'freshness' mechanism, namely: nonces (random challenges), timestamps, and sequence numbers. They each have their own advantages and disadvantages.

Most notably, the use of nonces typically requires an extra message to be sent, although it avoids the need for any long-term stored state information. On the other hand, the use of timestamps and sequence numbers, whilst typically involving one less message, require the entities to possess synchronised clocks or store long-term state information, respectively. It is this latter issue, i.e. the storage of state information and the problems that can arise in the event of a database failure, that is the main focus of this paper.

3 Server to client authentication

There are many situations where a single server is required to engage in authentication (and key establishment) exchanges with a potentially large number of clients. One particular example of this is provided by mobile telecommunications networks, where the server is the 'Service Provider' (SP) (or 'Home Network') and the clients are mobiles associated with this SP. For example, in GSM mobile telecommunications (see [4] for an overview) the mobile user is required to authenticate itself to the network from which it requires service. In this case the authentication is unidirectional, i.e. the mobile authenticates itself to the network and not vice versa, and timeliness is achieved by the use of nonces. More specifically, the mobile user's home network provides a set of authentication 'triples' to the 'visited network', where each triple consists of a random challenge, the response to the challenge, and a session key to be used to encrypt data subsequently exchanged between the mobile and the network. Note that both the response to the challenge and the session key are computed as a function of the random challenge and a secret key shared by the mobile and its SP.

The use of these triples has a number of operational advantages. By providing several of these at once to the visited network, the visited network is able to authenticate the mobile several times without further reference back to the SP, reducing the on-line load on the SP. Moreover, it means that the key shared by the mobile and its SP is not divulged to any visited network, and also the algorithms used to generate responses to challenges and session keys can be specific to an SP and need not be shared with the visited network (making upgrades to algorithms very simple).

Whilst unilateral authentication was deemed sufficient for GSM, in future networks mutual authentication is required to help protect against new threats. Using sequence numbers, the GSM authentication protocol can be enhanced in a simple way to provide authentication of the network to the mobile. This is achieved by making the GSM 'triples' into 'quadruples', through the inclusion of a sequence number with the challenge, response, and key. In addition the challenge incorporates a MAC computed as a function of both the sequence number and the key shared by the mobile and its SP.

This sequence number enhanced version of GSM authentication is included in the latest draft

3GPP security architecture, [1], based on a Siemens submission. In its simplest and 'most obvious' incarnation, this requires the mobile's SP and the mobile to each maintain a counter, which is used to generate and check the sequence number, respectively. The mobile will only accept and respond to a challenge if the sequence number is greater than any previously received.

What makes this solution particularly attractive is that it achieves the desired objective, i.e. authentication of the network, without adding any messages to the protocol (and with minimum additions to the existing messages). It also avoids any need for synchronised clocks, which are an unacceptable requirement in the mobile telecommunications environment.

4 Robustness and a timestamp solution

Despite the apparent attractiveness of the use of sequence numbers, there is a major problem with the use of the above 'obvious' method for sequence number generation and checking which is potentially serious enough to prevent them being used. This relates to the fact that both parties need to maintain counters to generate and verify sequence numbers, and, in particular, the SP needs to maintain a database of sequence numbers, one for each mobile. In the event of a database failure at the SP, a routine back-up of this database will not be sufficient to restore communications, at least for all those mobiles for which authentications have been performed since the back-up was created.

Of course, it would be possible to generate a log of all transactions, and use this in combination with routine back-ups to restore the database to its condition immediately prior to the failure. However, this is potentially both expensive and the process could itself fail. Providing one solution to this problem is the motivation for the main idea presented in this paper.

In solving this problem it is important to observe that counters are by no means the only way to generate sequence numbers. The following description of sequence numbers is taken from Annex B of [2].

> A claimant and verifier agree beforehand on a policy for numbering messages in a particular manner, the general idea being that a message with a particular number will be accepted only once (or only once within a specified time period). Messages received by a verifier are then checked to see that the number sent along with the message is acceptable according to the agreed policy.

Thus it would be perfectly acceptable for the SP to take its sequence numbers from a clock. The mobiles will only be required to keep the last received sequence number (a timestamp), and will accept a new sequence number if any only if it is 'more recent' than the stored value — the mobiles will certainly not need to be equipped with a clock.

This approach avoids any need for a back-up, since if the SP fails then it can simply be restarted with the correct time, which will guarantee that sequence numbers will be generated in a 'monotonic' fashion. Note further that a 32-bit timestamp value, counting seconds from some notional start date, offers over 136 years of use as a sequence number generator. Of course, this means that authentications can only be carried out once per second, but this is perfectly adequate for many applications, including the 3GPP application referred to above. Indeed,

this solution, as proposed by the author, is included in the latest version of the 3GPP Security Architecture, [1].

Finally note that, although the idea has been described in the context of mobile telecommunications, it is of general applicability, especially in systems where a single server authenticates itself to a large number of clients.

5 Conclusions

This paper has described the use of a clock to generate sequence numbers for use in authentication, usable even when the recipient does not have a synchronised clock. This scheme is robust, in the sense that it will continue to operate even if the sequence number sender loses its stored state. It is hoped that this contribution will be of value in making sequence number based authentication schemes more usable in practice.

References

[1] 3rd Generation Partnership Project (3GPP). *3G TS 33.102 version 3.0.0 — Technical Specification Group Services and System Aspects; 3G Security; Security Architecture*, May 1999.

[2] International Organization for Standardization, Genève, Switzerland. *ISO/IEC 9798–1: 1997, Information Technology — Security techniques — Entity authentication — Part 1: General*, Second edition, August 1997.

[3] L. Lamport. Time, clocks, and the ordering of events in a distributed system. *Communications of the ACM*, **21**:558–565, 1978.

[4] K. Vedder. Security aspects of mobile communications. In R. Govaerts B. Preneel and J. Vandewalle, editors, *Computer security and Industrial Cryptography: State of the Art and Evolution*, number 741 in Lecture Notes in Computer Science, pages 193–210. Springer-Verlag, Berlin, 1993.