Near Optimal Unconditionally Secure Authentication

Richard Taylor
Telematic and System Security
Telecom Australia Research Laboratories
P. O. Box 249
Clayton Victoria 3168
Australia

Abstract. An efficient unconditionally secure authentication scheme with arbitration is presented which is also secure against attacks by the arbiter. Arguments are presented that suggest this scheme is almost optimal with respect to codeword lengths, and when many messages are sent the amount of key data per message approaches, at worst, 1/3 more than the best possible. An improved unconditionally secure authentication scheme without arbitration is also briefly presented.

1 Introduction

Authentication in message transmission means that the receiver can reliably identify the sender and be confident that the message has not been altered or substituted in transit. Well known examples of authentication schemes are the RSA signature scheme [5], and the use of block ciphers, like DES, to provide integrity check values [3]. However the level of security these schemes actually provide have not been proven. In the case of the RSA scheme the level of security depends on the difficulty of factoring large integers. However, the difficulty of factoring remains unknown. Block ciphers, by their very nature, have properties that make integrity check functions that use them difficult to analyse precisely. In contrast unconditionally secure authentication schemes (sometimes called authentication codes) have a level of security that does not depend on any unproven assumptions (see [2], [9], and the survey article [8]).

For a given probability of successful attack the efficiency of an unconditionally secure authentication scheme may be considered in terms of the computations required by sender and receiver, the amount of shared key, and the length of codewords used to convey source messages. The following scheme is closely related to that of [9] but

requires about 1/4 of the key data and 1/2 of the authentication computations while maintaining the same codeword lengths.

2 Improved Unconditionally Secure Authentication

Let p be a prime number. Let a message M be divided up into w bit words m_1, m_2, ... , m_n, such that $2^w < p$. It is suggested that p be chosen to be close to a power of 2 for efficient calculation of products modulo p (see [4]). For example $p = 2^{31}-1$ and w = 30 are suitable values. Let a_1, a_2, ... , a_{j+2} where $j = \lceil \log_2(n) \rceil$, be integers modulo p that form a secret shared key between a sender and receiver. The authentication function F of the message M is defined below. The sequences s_0, s_1, ... , s_j are initialised by $s_0 = M$ and defined recursively in a way that approximately halves the length of successive s_i. All the arithmetic below is modulo p, and the value of F is in the range 0 to p–1.

$$s_0 = (m_1, m_2, m_3, \ldots, m_n).$$

If $s_i = (r_1, r_2, r_3, \ldots, r_t)$ *define*

$$s_{i+1} = \left\{ \begin{matrix} (a_{i+1}r_1 + r_2,\ a_{i+1}r_3 + r_4, \ldots,\ a_{i+1}r_{t-1} + r_t) & t \text{ even}, \\ (a_{i+1}r_1 + r_2,\ a_{i+1}r_3 + r_4, \ldots,\ a_{i+1}r_{t-2} + r_{t-1},\ r_t) & t \text{ odd}. \end{matrix} \right\}$$

Let $s_j = (v)$. *Then*

$$F(M, p, a_1, a_2, \ldots, a_{j+1}, a_{j+2}) = a_{j+1}v + a_{j+2}.$$

In this scheme the value of the authentication function F is simply appended to the message M and sent with it. Thus the authentication function is used like a message authentication code (mac) or integrity check value (icv). The theorem below indicates the strength of the integrity mechanism in terms of the likelihood of replacing, in transit, a message and the corresponding icv with a legitimate, but different, message-icv pair. The proof of the theorem is omitted as it is very similar to the proof of a similar result of [9] (see p 272).

Theorem. Let M and M' be any two unequal message strings of n words and y, g any fixed integers. If $a_1, a_2, \ldots, a_{j+2}$ *where* $j = \lceil \log_2(n) \rceil$, *are independent and uniformly distributed random numbers modulo p,*

$$Probability[F(M', p, a_1, a_2, \ldots, a_{j+1}, a_{j+2}) \equiv y \ (mod\ p)$$
$$/\ F(M, p, a_1, a_2, \ldots, a_{j+1}, a_{j+2}) \equiv g \ (mod\ p)] \leq \frac{\lceil log_2(n) \rceil}{p}.$$

Note that the amount of key required in the calculation of F is at most $\log_2(n)+3$ integers modulo p or $(\log_2(n)+3)\log_2(p)$ bits. Also the calculation of F requires at most n multiplications modulo p and n additions modulo p. In any multiplication based scheme this is probably the fewest number of multiplications possible since presumably every block of message needs to be multiplied at least once by the authentication function. In comparison the scheme suggested in [9] requires about $4\log_2(n)\log_2(p)$ bits of key, and involves approximately 2n multiplications and 3n additions modulo p. In both the scheme of [9] and the one presented here many messages may be sent by re-using the same $a_1, a_2, \ldots, a_{j+1}$ and using a new value of a_{j+2} for each message (see[9]). This gives an average amount of key approaching just one integer modulo p, or $\log_2(p)$ bits, per message. Nevertheless, the computational advantage of the scheme given here remains.

3 Improved Authentication With Arbitration

Although authentication schemes protect against attacks from outsiders, they may not protect against misuse by the sender or receiver. For example, having sent a message, the sender may later wish to deny having sent it. Or the receiver of a message may wish to alter or replace a legitimately received message, and claim it to be authentic. In this situation a dispute may arise between the sender and receiver. In [6] (see also the full paper [7]) a solution to this problem is provided with the participation of a third party called an arbiter. However as the author points out the arbiter can impersonate the sender in a way that the receiver will not detect. This problem is eliminated in the scheme of [1], but this scheme is not nearly as efficient as that of [9]. The scheme presented below answers a question of [1] by providing an arbitrated scheme with comparable efficiency to that of [9].

Let a sender, receiver, arbiter and some hostile outsider be denoted by Sally, Ray, Alice, and Oliver, respectively. Sally wishes to send messages to Ray and for this communication to be unconditionally secure against the following attacks:

Attack 1. Alice or Oliver generates a message, or alters one in transit, and attempts to send this to Ray as if it came from Sally. In this attack Ray or Sally do not defer to Alice.

Attack 2. Ray generates a message, or alters one received from Sally, and attempts to claim that it was sent by Sally. In this case Alice is defered to in an attempt to detect this attack.

Attack 3. Sally sends a message that is accepted by Ray as coming from Sally, and later attempts to deny that the message was sent by her. As in Attack 2, Alice is defered to in an attempt to detect this attack.

The following scheme has similarities to that of [1] but contains important modifications that enhance the efficiency. As in [1] the scheme involves a number of phases. In the key sharing phase information is securely exchanged and certain calculations made by Ray, Sally, and Alice. In the transmission phase Sally sends one or more messages to Ray, Ray receives these messages and subjects them to a verification procedure designed to verify their authenticity. It is important for practical reasons that Alice has no involvement in the transmission phase. Finally in case of a dispute of Type 2 or 3, Alice is requested to resolve the situation.

As before, a message M is represented in terms of w-bit words $m_1, m_2, \ldots, m_n$, and a prime number $p > 2^w$ is chosen. In the scheme described sufficient key is exchanged among Ray, Sally and Alice to allow Sally to send Ray any t messages, each consisting of n words of w bits. The essential notion behind the construction is the use of hyperplanes in n+t+3 space with arithmetic over the field modulo p. These hyperplanes are specified by n+t+3 numbers (not all zero) $r_1, r_2, \ldots, r_n, s_1, s_2, \ldots, s_t$, a, b, c, between 0 and p–1, corresponding to the n+t+2 dimensional hyperplane $r_1v_1 + r_2v_2 + \ldots, + r_nv_n + s_1w_1 + s_2w_2 + \ldots + s_tw_t + ax + by + cz = 1$, with axes $v_1, v_2, \ldots, v_n, w_1, w_2, \ldots, w_t$, x, y, z. This hyperplane will also be expressed as an n+t+3 tuple in square brackets, ie. $[r_1, r_2, \ldots, r_n, s_1, s_2, \ldots, s_t, a, b, c]$. For a hyperplane P we shall use [P] to refer to the corresponding n+t+3 tuple. Note that throughout the remainder of this paper all the arithmetic and equations are assumed to be over the field modulo p.

Key Sharing

Step 1. Ray randomly selects the hyperplane $[P_R] = [r_1, r_2, \ldots, r_n, s_1, s_2, \ldots, s_t, a, b, c]$, by selecting the coordinates uniformly and independently from the integers 0 to p−1. In the event that $b = 0$, b is randomly re-chosen until $b \neq 0$. Ray then randomly selects another hyperplane $[P_S] = [r_1', r_2', \ldots, r_n', s_1', s_2', \ldots, s_t', a', b', c']$. It is required that $c' \neq b^{-1}b'c$. If this is not so c' is randomly re-chosen until the condition is met. The condition $c' \neq b^{-1}b'c$ ensures that P_R and P_S are not parallel. Ray secretly shares P_S with Sally and P_R with Alice.

Step 2. Alice randomly selects a hyperplane $[P_A] = [r_1'', r_2'', \ldots, r_n'', s_1'', s_2'', \ldots, s_t'', a'', b'', c'']$. If any of the conditions $b'' \neq b$, $c'' \neq c$, and $c'' \neq b^{-1}b''c$, do not hold then b" and c" are randomly re-chosen until they do. The condition $c'' \neq b^{-1}b''c$ ensures that P_R and P_A are not parallel. Alice then forms the n+t+1 dimensional hyperplane $P_R \cap P_A$. It is desirable to express $P_R \cap P_A$ in such a way that $P_R \cap P_A$ may be conveyed but the individual hyperplanes P_R and P_A not revealed. This may be done by expressing $P_R \cap P_A$ as the intersection of two planes parallel with the y and z axes respectively. Thus $P_{R'}$ and $P_{A'}$ are calculated in which $P_{R'} \cap P_{A'} = P_R \cap P_A$ and where $[P_{R'}] = [g_1, g_2, \ldots, g_n, h_1, h_2, \ldots, h_t, d, 0, f]$ and $[P_{A'}] = [g'_1, g'_2, \ldots, g'_n, h'_1, h'_2, \ldots, h'_t, d', e', 0]$. This representation is ensured possible by the conditions $b'' \neq b$ and $c'' \neq c$. Alice secretly shares $P_{R'}$ and $P_{A'}$ with Sally, so that she can determine $P_{R'} \cap P_{A'}$.

Step 3. Sally checks that

$$Determinant \begin{bmatrix} d & 0 & f \\ d' & e' & 0 \\ a' & b' & c' \end{bmatrix} \neq 0.$$

If this is not so Sally requests Alice to make a new random choice of a" and to repeat Step 2 based on the new P_A formed. This will ensure that the determinant above is non-zero, and that P_S and $P_{R'} \cap P_{A'}$ are not parallel. Note that if Sally requests Alice to make a new choice of a" then Alice will know information about a', b' and c', however this cannot be usefully used by Alice in an attack (see Section 4). Sally forms the n+t dimensional hyperplane $P_{R'} \cap P_{A'} \cap P_S$.

After the key sharing has been completed each of the parties Oliver, Ray, Sally, and Alice has incomplete information about the hyperplanes P_R, P_S and P_A. Oliver does not know P_R, P_S or P_A; Ray knows P_R and P_S but not P_A; Sally knows P_S and $P_{R'} \cap P_{A'}$ (= $P_R \cap P_A$) but not P_R or P_A; Alice knows P_R and P_A but not P_S.

Transmission

To send the message $M_1 = m_1^1, m_2^1, \ldots, m_n^1$ Sally calculates the point p_1 on the hyperplane $P_R \cap P_A \cap P_S$ of the form $(m_1^1, m_2^1, \ldots, m_n^1, 1, 0, 0, \ldots, 0, x^1, y^1, z^1)$. There will be exactly one such point. Sally sends p_1 to Ray (only the n+3 tuple $(m_1^1, m_2^1, \ldots, m_n^1, x^1, y^1, z^1)$ need actually be sent). Ray verifies that the message is from Sally by checking that p_1 is on $P_R \cap P_S$. To send the uth message ($u \leq t$), $M_u = m_1^u, m_2^u, \ldots, m_n^u$, Sally calculates the point p_u on the hyperplane $P_R \cap P_A \cap P_S$ of the form $(m_1^u, m_2^u, \ldots, m_n^u, 0, 0, \ldots, 0, 1, 0, 0, \ldots, 0, x^u, y^u, z^u)$, with a 1 in the $n+u^{th}$ coordinate. Sally sends p_u to Ray. Ray verifies that the message is from Sally by checking that p_u is on $P_R \cap P_S$. Note that the points p_1, p_2, ..., p_t generated are linearly independent.

Arbitration

In case of a dispute Ray takes p_i to Alice. Alice checks whether p_i lies on $P_R \cap P_A$. If so p_i is deemed to have been sent by Sally.

4 Analysis

Attack Probabilities

Theorem. The probabilities of successful attacks of Types 1 - 3 are all bounded by $1/(p-1)$.

Proof. In an attack of Type 1 it is sufficient to consider the case where the attacker has the maximum amount of information available on which to base an attack. This is the Type 1 attack as performed by Alice in which a message is altered in transit. As described in the transmission phase, assume that messages M_1, M_2, ..., M_{u-1}, where $u-1 < t$, have been sent, received and validated. Further assume that Alice has read the corresponding points p_1, p_2, ..., p_{u-1}. Also assume that Alice has intercepted the point $p_u = (m_1^u, m_2^u, \ldots, m_n^u, 0, 0, \ldots, 0, 1, 0, 0, \ldots, 0, x^u, y^u,$

z^u) associated with the message M_u and replaced it with the point $p^* = (m_1^*, m_2^*, \ldots, m_n^*, 0, 0, \ldots, 0, 1, 0, 0, \ldots, 0, x^*, y^*, z^*)$ associated with a different message M^*. As $M^* \neq M_u$ there must be some j with $m_j^* \neq m_j^u$. For the message M^* to be accepted by Ray the point p^* must be on the hyperplane P_S. The information Alice has about P_S is embodied in the inequality $c' \neq b^{-1}b'c$ (note that Alice knows P_R and therefore b and c), and the knowledge that $p_1, p_2, \ldots, p_u$ are on P_S. The latter conditions may be summarised by the equations

$$\begin{aligned} &r_1'm_1^1+\ldots+r_n'm_n^1 + s_1' + a'x^1 + b'y^1 + c'z^1 = 1 \\ &\cdot \\ &\cdot \\ &\cdot \\ &r_1'm_1^u+\ldots+r_n'm_n^u + s_u' + a'x^u + b'y^u + c'z^u = 1. \end{aligned} \qquad (1)$$

These equations may be rewritten as

$$\begin{aligned} &s_1' = 1 - r_1'm_1^1-\ldots-r_n'm_n^1 - a'x^1 - b'y^1 - c'z^1 \\ &\cdot \\ &\cdot \\ &\cdot \\ &s_u' = 1 - r_1'm_1^u-\ldots-r_n'm_n^u - a'x^u - b'y^u - c'z^u. \end{aligned} \qquad (2)$$

It is clear that these equations place no restrictions on the collection of unknowns $\{r_1', r_2', \ldots, r_n', a', b', c'\}$ as $s_1', s_2', \ldots, s_u'$ are uniformly distributed independent random variables. Consider the expression

$$r_1'm_1^*+\ldots+r_n'm_n^* + s_u' + a'x^* + b'y^* + c'z^*. \qquad (3)$$

This must be 1 if M^* is to be accepted by Ray. Substituting for s_u' from the last equation of (2) into (3) gives

$$\begin{aligned} &r_1'(m_1^* - m_1^u)+\ldots+r_n'(m_n^* - m_n^u) + \\ &\qquad a'(x^* - x^u) + b'(y^* - y^u) + c'(z^* - z^u) + 1. \end{aligned} \qquad (4)$$

Since $m_j^* - m_j^u \neq 0$, then the fact that r_j' is independent of $r_1', r_2', \ldots, r_{j-1}', r_{j+1}', \ldots, r_n', a', b', c'$ means that (4) has a value equally distributed among the numbers 0,

..., p−1. Thus the probability that Ray will accept p*(and so M*) will be at most 1/p. Attack 2 may be analysed similarly, and the details are omitted. In this case Ray does not know P_A but must form a point on P_A that is different from any point previously sent by Sally.

In Attack 3, Sally sends a message that is accepted by Ray as coming from Sally, and later attempts to deny that the message was sent by her. For this attack to succeed Sally must form a point on P_R and P_S which is not on P_A. As in the analysis of Attack 1, assume that u messages have been sent received and validated and that p* = (m_1*, m_2*, ... , m_n*, 0, 0, ..., 0, 1, 0, 0, ..., 0, x*, y*, z*) is such a candidate point. It is straightforward to show that the collection of possible hyperplanes P_R that may correspond to P_A' and P_R' may be expressed by the linear forms

$$(1-t)[P_{A'}] + t[P_{R'}], \text{ where } t \neq 0 \text{ (from } b \neq 0). \qquad (5)$$

Consider the expressions

$$g_1 m_1'' + \ldots + g_n m_n'' + h_u + dx'' + fz'', \qquad (6)$$

$$g'_1 m_1'' + \ldots + g'_n m_n'' + h'_u + d'x'' + e'y''. \qquad (7)$$

Let (6) and (7) have the values q_1 and q_2 respectively. If p* is not on $P_R \cap P_A = P_{R'} \cap P_{A'}$ then either $q_1 \neq 1$ or $q_2 \neq 1$. If p* is on $P_R \cap P_S$ and therefore on P_R then combining (5), (6) and (7)

$$(1-t)q_1 + tq_2 = 1. \qquad (8)$$

If $q_2 \neq 1$ then from (8) $q_1 - q_2 \neq 0$, and (8) may be written as $t = (1-q_2)/(q_1-q_2)$. Given that t may take any value modulo p except 0 (from (5)), there is a probability of at most 1/(p−1) that (8) will be satisfied, and that Ray will accept p*. If $q_2 = 1$ and $q_1 \neq 1$, then (8) cannot hold (since $t \neq 1$) and so Ray will not accept p*. This completes the proof. //.

5 Efficiency

The length of codewords is $nw+3\log_2(p)$ which is just $3\log_2(p)$ bits longer than the source messages they convey. The amount of key information shared is

$(4(n+t)+10)\log_2(p)$ which allows for t messages to be sent. For large t this tends to $4\log_2(p)$ bits of key per message. On the other hand in a simple unconditionally secure authentication channel (without arbitration) codewords must convey the message as well as contain the result of any one of 1/P authentication functions (where the probability of a successful attack is at most P). It is not dificult to see that this requires codewords of length at least $nw+\log_2(1/P)$ bits (see [8] for example). The sender and receiver must also agree on one of 1/P authentication functions, which requires at least $\log_2(1/P)$ bits of shared key. Now in any arbitrated authentication scheme as described here each message from Sally to Ray must also (and independently) be an unconditionally secure communication from Ray to Alice (for Type 3 attacks) and from Sally to Alice (for Type 2 attacks). It would appear then that the average length of a codeword must be at least $nw+3\log_2(1/P)$, and the amount of key data shared at least $3\log_2(1/P)$.

Since $\log_2(1/P) > \log_2(p-1)$ the scheme presented is essentially optimal with respect to codeword lengths (within $3\log_2(p)-3\log_2(p-1)$, which tends to 0 for large p). From the argument sketched above it would appear that when many messages are sent the amount of key data per message used approaches, at most, 1/3 more than that required by any such system.

In comparison the scheme of [1] uses an amount of shared key data proportional to $n\log_2(p)$, and codewords of length four times that of the source messages conveyed (or 4nw). Furthermore it is not well suited to sending long messages as it involves the multiplication of integers that are as long as the messages.

Acknowledgement

The author wishes to thank Edward Zuk for discussions concerning the work in this paper, and his valuable comments on an early draft. Also the permission of the Director, Research, of Telecom Australia to publish this paper is hereby acknowledged.

References

[1] Y. Desmedt and M. Yung. Arbitrated Unconditionally Secure Authentication Can Be Unconditionally Protected against Arbiter's Attacks. Advances in Cryptology - CRYPTO '90, proceedings, Springer-Verlag 1991, pp. 177-188.

[2] E. Gilbert, F. MacWilliams, and N. Sloane. Codes which detect deception. The BELL System Technical Journal, 53(3), pp. 405-424, March 1974.

[3] ISO/IEC 9797. Data cryptographic techniques-Data integrity mechanism using a cryptographic check function employing a block cipher algorithm, International Organisation for Standardisation, 1989.

[4] H. J. Knobloch. A Smart Card Implementation of the Fiat-Shamir Identification Scheme, Advances in Cryptology-EUROCRYPT '88, proceedings, Springer-Verlag 1989, pp. 87-96.

[5] R. L. Rivest, A. Shamir, and L. Adleman. A method of obtaining digital Signatures and public key cryptosystems. Commun. ACM, 21, pp. 294-299, April 1978.

[6] G. J. Simmons. Message authentication with arbitration of transmitter/receiver disputes. Advances in Cryptology - EUROCRYPT '87, proceedings, Springer-Verlag 1988, pp. 151-165.

[7] G. J. Simmons. A cartesian product construction for unconditionally secure authentication codes that permit arbitration. Journal of Cryptology, 2(2), pp. 77-104, 1990.

[8] G. J. Simmons. A survey of information Authentication, Contemporary Cryptology - The Science of Information Integrity, 1991, pp. 379-419. IEEE Press, New York.

[9] 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, 1981, pp. 265-279.