ZERO-KNOWLEDGE PROOFS OF IDENTITY AND VERACITY OF TRANSACTION RECEIPTS

Gustavus J. Simmons[a] and George B. Purdy[b]

[a] Sandia National Laboratories
Albuquerque, NM 87185

[b] University of Cincinnati
Department of Mathematics
Cincinnati, OH 45221

Abstract

There are two equally important, related, functions involved in the control of assets and resources. One of these is the verification of a potential user's identity and authority to use or have access to those assets. The other is to provide a record (receipt) of each access so that in the event of a later dispute as to whether an illegitimate use was made of the assets, or of the extent of the liability incurred in a legitimate use, etc., the authenticity and specifics of the access can be demonstrated in a logically compelling (and hence eventually legally binding) manner to an impartial third party or arbiter. Elaborate, and legally accepted, document based protocols to accomplish these functions are central to all commercial and private transactions. When the resources are remotely accessible, however, as in the case of computer data files, electronic funds transfers (EFT), automated bank tellers, and even in many manned point-of-sale systems, no satisfactory counterpart to the established document based protocols for verifying individual identity and/or authority to use a resource have been found, nor has a fully satisfactory means been devised to provide unforgeable transaction receipts. In this paper, we show how a public authentication channel can be used to certify private (user unique) authentication channels in a protocol that both "proves" a potential user's identity and authority and also provides certified receipts for transactions whose legitimacy can later be verified by impartial arbiters who did not have to be parties to the original transaction.

We also introduce an authentication scheme to be used in this application based on the legitimate originator of information being able to extract square roots modulo $n = pq$, where p and q are primes of a special form. We show that these protocols provide a zero-knowledge proof of identity and of veracity transaction receipts, and that they are therefore very secure. We also show how the legitimate owner of the authentication channel can give a zero-knowledge proof that the modulus

[a] This author's work performed at Sandia National Laboratories supported by the U. S. Department of Energy under contract no. DE-AC04-76DP00789.

C.G. Guenther (Ed.): Advances in Cryptology - EUROCRYPT '88, LNCS 330, pp. 35-49, 1988.

n has the correct form, thereby eliminating the possibility of the existence of several known subliminal channels.

Introduction

There are two parts to the problem of verifying the identity of an individual whom we will refer to as the user, whether remotely or face-to-face. First, the party or device making the identification (the verifier) must have identifying information available to match or check against the information submitted to support a claimed identity. Clearly, the confidence that the verifier has in any particular identification can be no greater than his confidence in the integrity of the corroborating information on which the identification is based. Consequently, the first part of the identity verification problem is to devise means by which the verifier can have access to identifying information whose integrity he can trust. This information may either be intrinsic to the individual being identified, such as physiognomy, fingerprints, voice prints, retinal prints, dynamics of a written signature, etc., or else it may be extrinsic, i.e., a private (secret) piece of information such as a computer access password, a telephone credit card number, a personal identification number (PIN), etc., not intrinsically associated with the individual, but whose possession is equated with the user's identity. The second part of the identity verification problem for extrinsic identification is to devise means to protect this identifying information from forgery or fraudulent use, especially to insure that as a consequence of someone eavesdropping on repeated uses by the legitimate user that they cannot improve their chances of impersonating him. Assuming that there are many users whom a verifier may have to identify, the file of identifying information that he uses for this purpose may take the form of an actual trusted directory, perhaps hidden behind a one-way function [8,12,20] to protect the users against the verifier or his agents impersonating them to other verifiers, or it may be an implicit directory in which the user produces trusted (?) identification credentials, such as drivers licenses, photo ID's, major credit cards, etc., in support of his access request at the time it is made. It should be pointed out that in transactions where significant liability is involved, these user supplied credentials are often themselves verified by querying a central file; telephone verification of credit cards at the point of sale, etc. This defeats the main purpose of having user-supplied means of identification, i.e., to make identification a purely local protocol, but is made necessary by the low level of confidence achievable in conventional user-supplied means of identification. In either case, whether the directory is actually in the possession of the verifier or is merely remotely accessible by him, trust in the directory is derived from trust in the integrity of the issuer of the directory.

In the first reported application of public key crypto techniques (fielded by the Sandia National Laboratories in 1978), an authentication channel based on the

RSA cryptoalgorithm was used to create trusted credentials that users could carry with them and present to the verifier at the time they requested access, in this case to the very sensitive Zero Power Plutonium Reactor at Idaho Falls, Idaho [7,16]. The public authentication channel (a publicly known RSA modulus n and decryption exponent d) was used by the issuing office of the Atomic Energy Commission to authenticate (certify) a text that included physical descriptors for the individual being identified is well as the details of the nature, type, duration, etc., of the access authorized. The object of this scheme was to make it possible for each user to carry with him what would have effectively been his entry in the verifier's trusted directory (a trusted credential in this case), that could be authenticated by the verifier, but which would be of no assistance to anyone wishing to produce a fraudulent credential. In this particular application, the identification information was intrinsic to the user (hand geometry, body weight, etc.), however, in other applications [16] the same basic technique has been used with extrinsic information in a manner similar to the protocol to be described here.

The essential concept in the protocol to provide verifiable proof of identity and unforgeable certified receipts is to use a public authentication channel to create trusted credentials which users will keep in their possession which certify, along with various identifying information, the public part of a user-unique authentication channel: the private (secret) part of which is known only to the legitimate user identified in the credential [19]. These credentials need not be kept secret and consequently avoid the necessity of generating, distributing and protecting local trusted directories or of establishing secure communications (authentication) channels to permit access by the verifiers to centralized trusted directories. At the time a user presents a credential (not necessarily his own) the verifier can first establish locally, via the public authentication channel that the credential is valid, i.e., that it was created by the issuer, and secondly, that the user identified in the now authenticated credential knows the private part of an authentication channel whose public part is described there. The applicant can then "prove" (in probability) that he is the individual to whom that credential belongs by demonstrating that he can authenticate challenge messages submitted by the verifier whose authenticity the verifier can establish using the (certified) public part of the authentication channel described in the credential.

The Protocol

The protocol described here presupposes the existence of an unconditionally trusted issuer of validated (signed) identification credentials. This could be a government agency, a credit card center or financial institution, a military command center, a centralized computer facility, etc. The issuer first establishes a public authentication channel to which he retains the secret authenticating function. As mentioned earlier, this could be any suitably secure authentication channel. The one we will use to illustrate the protocol is based on the computational equivalence (in probability) of extracting modular square roots and of factoring a composite modulus. To set up such a channel, the issuer first chooses a pair of primes p and q; $p \equiv 3 \pmod 8$ and $q \equiv 7 \pmod 8$. p and q must satisfy the same conditions required to construct a "good" RSA modulus, i.e., p and q must be chosen so that it is computationally infeasible for anyone to factor the modulus $n = pq$. There are two reasons for requiring that $p \equiv 3 \pmod 8$ and $q \equiv 7 \pmod 8$. The first, which is simple to explain, is to make it easy for anyone who knows the factors to extract the modular square root of a square with respect to n.[1] The second reason is harder to explain in detail, but basically it is to guarantee that there is a unique, but publicly determinable, square associated with every message, u, that may need to be authenticated. The explanation of why we want this to be true we will defer for the moment. This restriction on the choice of p and q represents no significant increase in the computational difficulty of finding suitable primes during the initial set up of the authentication channel. The issuer keeps the factorization of n secret; in fact, the security of the system against fraudulent claims of validated identity is no better than the lesser of

a) the quality of protection provided p and q by the issuer

or,

b) the difficulty of factoring n.

The issuer must also have available a polyrandom function f that maps arbitrary strings of symbols to the range [0,n). By polyrandom, we mean that f cannot be distinguished from a truly random function by any polynomially bounded computation. f will be a publicly known function, and need not change over the lifetime of the identification protocol. Many strong, single-key cryptographic functions, such as the DES when used with a fixed publicly known key in a block chain encryption mode, appear to adequately approximate this condition. n and f are the public part of the issuer's authentication channel. The private (secret) part of the channel, known only to the issuer, is his knowledge of the factors p and q. Since taking modular square roots is computationally equivalent (in probability) to factoring n, the issuer can prove that he is who he claims to be, i.e., prove that he knows the factorization of n, by being able to produce square roots modulo n. The issuer cannot simply authenticate arbitrary messages submitted to him by public receivers

[1] Given a prime p and a quadratic residue, y, of p it is only an O(log p) computational task to find a solution to the quadratic congruence

(i) $$x^2 \equiv y \pmod p \quad ,$$

i.e., to extract a modular square root of y. This is true irrespective of the choice of the prime p, however if $p \equiv 3 \pmod 4$ the solution of (i) is particularly simple:

(ii) $$x \equiv \pm\, y^{(p+1)/4} \pmod p$$

where the - indicates the complement (mod p). Exponentiation is only an O(log p) computational task using the well-known square-and-multiply algorithm [6].

(either users or verifiers), since each time he responded with a square root to a square chosen by someone else he would potentially compromise the factorization of n, and hence the capability to fraudulently authenticate messages in his stead, with probability 1/2. Similarly, a receiver can't accept an arbitrary square and matching square root as proof of the identity of the party possessing them, since anyone could choose an arbitrary x and square it to calculate a matching square with respect to the issuer's publicly known modulus, n. Consequently, the squares that the issuer will authenticate, i.e., whose square roots he will extract, must be indeterminate to both the issuer and the receiver in order for the public authentication channel to be secure; both against the receiver being deceived as to the identity of the originator of a message and to the issuer against having his identity usurped. The primary purpose of the polyrandom function f is to provide this indeterminacy. It's secondary purpose is to map strings of symbols (whose length may vary) into the range $[0,n)$, i.e., into the principal residues of n.

In the usual communications usage of an authentication channel, a transmitter wishes to send a message, m, to public receivers and to "prove" to them that the communication came from him and not from someone impersonating him, and also that a message hasn't been altered after he signed it. To do this with the authentication channel just described, the transmitter would, if necessary, introduce additional redundant information, typically a field of the message filled with a publicly known symbol, say a terminal block of k zeros, to form an extended message, $\bar{m}$. $\bar{m}$ will be a square modulo n with probability 1/4, in which case the transmitter can extract a square root, s, and send the couplet (m,s) as the authenticated (signed) message. There are four square roots for m modulo n, one of which is chosen with a uniform probability distribution. The computational algorithm (modular square root) takes care of this random choice automatically. The transmitter need only communicate the message, m, not the extended message, $\bar{m}$, since the redundant information is publicly known so that the receiver can construct $\bar{m}$ from m in the same way that the transmitter did. The receiver(s) will accept (m,s) as an authentic communication from the transmitter if and only if

$$\bar{m} \equiv s^2 \pmod{n} \quad . \tag{1}$$

With probability 3/4, however, $\bar{m}$ will not be a square so that there is no s satisfying (1). In the case of a communications usage of the authentication channel, there are a variety of simple procedures by which the transmitter can cause the extended message $\bar{m}$ that he uses to be a square but, as we shall see, none of these are available in the present case since the transmitter must not be able to force the choice of the square to a value of his choice. In the identification protocol, the issuer would form the extended message $\bar{m}$ in exactly the same way the transmitter does in the communications example. But he would then form $u = f(\bar{m})$, depending on the polyrandom nature of f to protect himself from a compromise of the factorization of n that could occur if m was chosen (or could be sufficiently influenced) by the receiver and the receiver from deception by someone impersonating the issuer and presenting an arbitrary pair m and s satisfying (1), etc. If log(u) >> k, i.e., if the number of bits in u is much larger than k, then the probability of a randomly selected u actually being the image of some extended message with the proper k bits of redundant information will be 2^{-k}. The probability that u will be a square with respect to n is 1/4 as mentioned earlier, in which case the issuer can sign u by extracting the square root, etc. If u isn't a square, however, since f is a polyrandom function there is no evident way to manipulate $\bar{m}$ so as to cause u to become a square. In fact, if there were any way to influence the quadratic residuosity of u through f then f would not satisfy the definition of a polyrandom function, and the authentication channel would not be cryptosecure. Therefore, since it is computationally infeasible for the issuer to cause $u = f(\bar{m})$ to be a square, and since being able to extract modular square roots is the only means the issuer has of proving that he knows the factorization of n and hence of authenticating messages, we need a simple and publicly known, means of associating a unique, but publicly determinable square with u, for all residues u.

At this point, we remind the reader of two simple facts from elementary number theory: the product of either a pair of quadratic residues or of a pair of quadratic nonresidues is a quadratic residue, while the product of a quadratic residue with a quadratic nonresidue is a quadratic nonresidue. A quantity, u, $(u,n) = 1$, is a quadratic residue with respect to a composite modulus $n = pq$, if and only if it is a quadratic residue with respect to both p and q individually.

We also need two further number theoretic results [2]:

a) 2 is a quadratic residue of all primes of the form $P \equiv 1$ or 7 (mod 8) and a quadratic nonresidue if $P \equiv 3$ or 5 (mod 8).

b) -1 is a quadratic residue of all primes of the form $P \equiv 1$ (mod 4) and quadratic nonresidue if $P \equiv 3$ (mod 4).

The important thing to note is that 2 is a quadratic residue of q but is a quadratic nonresidue of p by (a) and that -1 is a quadratic nonresidue of both p and q by (b). This was why p and q were chosen to satisfy $p \equiv 3$ (mod 8) and $q \equiv 7$ (mod 8). Williams [22] was apparently the first to construct RSA moduli using primes of this special form which he exploited to resolve an ambiguity in the decryption of ciphers in a variant to the RSA cryptoalgorithm proposed by Rabin [14] for which they proved that decryption of (almost all) ciphers and of factoring the modulus were computationally equivalent.

Now consider an arbitrary residue u, $(u,n) = 1$. u can be classified into one of four classes according as to whether it is a quadratic residue or a quadratic nonresidue with respect to p and with respect to q. We represent these four classes as QR,QR; QR,NQR; NQR,QR and NQR,NQR; where the quadratic residuosity with respect to p is indicated first and with respect to q second. Now consider the classification of the four multipliers 1, -2, 2, -1: these are QR,QR; QR,NQR; NQR,QR and

NQR,NQR, respectively. Consequently, there will be precisely one quadratic residue (square) in the set of four residues

$$(2) \qquad (u,\ -2u,\ 2u,\ -u)$$

for any choice of a residue u, $(u,n) = 1$. The square residue is the product of u with the multiplier having the same classification as u. It is easy for the issuer to determine the class that u belongs to since he knows the factorization of n and hence easy for him to determine which of u, -u, 2u or -2u is a quadratic residue with respect to n. The issuer can therefore extract a (random) square root, s, of the unique quadratic residue associated with u and sign u with s. In the protocol described here, he also appends two additional bits b_2b_{-1} so that an authenticated message is of the form

$$(u;s:\ b_2b_{-1}) \quad ,$$

to inform whoever wishes to validate the authenticated message which one of the residues u, -2u, 2u or -u, respectively, he should expect to recover from the quadratic congruence,

$$(3)[2] \qquad s^2 = \ ?\ (\text{mod } n) \quad .$$

It isn't essential that the issuer append the two bits that tell which of the four cases to expect, since the verifier could compute t and then check to see whether t is one of u, -2u, 2u or u. If it is, then m would be accepted as an authentic message. It is simply computationally more efficient to append the two bits to the authenticated message than to have the verifier make the four tests. No extra information, i.e., no information not otherwise available, is conveyed by the appended pair of bits. By the convention used here (in arranging the entries in the array (2)), $b_2 = 1$ says multiply u by 2 while $b_{-1} = 1$ says to multiply by -1 to form the expected residue.

[2] The reader may recall a digital signature scheme proposed by Ong, Schnorr and Shamir [9,10] which superficially resembles the scheme described here. In their scheme, a composite modulus n and a residue k were made public. A signed message, m, was any triple (x,y;m) such that

$$(\text{i}) \qquad x^2 + ky^2 = m \qquad (\text{mod } n)$$

x and y were easy to calculate if one knew the factorization of n, but thought to be as hard as factoring otherwise. Pollard and Schnorr [11] have shown this not to be the case however. The problem is that in this signature scheme each message m has on the order of n signatures, i.e., pairs of integers x and y satisfying (i), hence it is computationally feasible to find some one out of these many pairs. In the scheme described here there is a unique signature for each message, so that the cryptographic weakness arising from having multiple signatures does not occur.

The probability that an opponent can find a u and s that satisfy (3) and have the required redundant information present in the preimage of u under f without knowing the factorization of n is 2^{-k} as has already been pointed out.

In the protocol, user i's identity is completely specified in an identifier (string of symbols), I_i, consisting of such information as his social security number, his bank account or credit card number, his military ID, etc., which could also include intrinsic physical descriptors, as well as any limitations on the authorization conveyed in the signed identifier, such as credit limits, expiration date, levels of access, etc. Most importantly, I_i must include the public part of the user's personal authentication channel consisting in the present example of an RSA modulus n_i, where $n_i = p_i q_i$ and $p_i \equiv 3 \pmod 8$ and $q_i \equiv 7 \pmod 8$ as required in setting up the issuer's public authentication channel; $n_i < n$. In addition, since anyone wishing to forge a credential could construct an identifier, I, to suit his purposes, I_i must include sufficiently much publicly known redundant information, such as message format, fixed fields of symbols common to all identifiers, I_i, etc, to make a forward search type attack [15] infeasible. The issuer first calculates

$$d_i = f(I_i) \quad . \tag{4}$$

and determines the classification of d_i according to its quadratic residuosity with respect to p and q. He then calculates the (least positive) square root of the unique quadratic residue associated with d_i. The authenticated (signed) credential

$$\left\{I_i ; s_i : (b_2 b_{-1})_i\right\}$$

is given to user i. No part of this credential need be kept secret. However, the user must keep secret his private authentication function: the factors p_i and q_i. His security against impersonation is totally dependent on him protecting this information, since his proof of identity in the scheme is equated to knowing the factorization of n_i.

The public part of the (issuer's) authentication channel is the issuer's modulus n, the polyrandom function f and a knowledge of the redundant information present in all of the I_i, which, as has been noted, must be sufficient to prevent a forward search cryptanalytic attack [15] on the polyrandom function f. In other words, the redundancy must be adequate to prevent someone wishing to fraudulently validate an identity from simply calculating $s_j^2 = t$ for randomly chosen signatures s_j until he finds a match with an $s_j = f(I)$ for some usable I -- this is the forward search attack. By making I contain sufficient redundant information, the probability of success of this sort of attack can be made as small as desired.

When user i wishes to prove his identity to a party A, say to gain access to a restricted facility or to log on to a computer or to withdraw money from an ATM, etc., he initiates the exchange by identifying himself to A using his identification credential and making his access request;

$$i \xrightarrow{\left[I_i;s_i:(b_2b_{-1})_i):t_j\right]} A \qquad \text{STEP 1}$$

t_j is a string of symbols that describes or identifies the transaction user i is requesting; t_j could be the date, the amount of the withdrawal, etc. A, who need not have an identification credential issued by the trusted issuer first verifies that the credential submitted to him is actually an authentic credential signed by the issuer. He accepts the credential (and the information contained in I_i) as genuine if and only if the quadratic congruence

$$(5) \qquad s_i^2 \equiv (2)^{b_2}(-1)^{b_{-2}}\left[f(I_i)\right] \qquad (\text{mod } n)$$

is satisfied. At this point in the protocol, if the test in (5) has been satisfied, A is confident that the credential $\left[I_i;s_i:(b_2b_{-1})_i\right]$ was issued by the issuer and that user i identified in I_i can authenticate messages using the private authentication channel described in I_i, in other words, for the example of an authentication channel being used here, that user i knows the factorization of n_i. The remaining question to A is whether the applicant who submitted the credential $\left[I_i;s_i:(b_pb_q)_i\right]$ is actually user i. This question can be answered by using the, now validated, private authentication channel.

A replies to the access request with a string of symbols, T_j, that describe the transaction from his standpoint; terminal ID, transaction number, confirmation of withdrawal amount, etc.

$$i \xleftarrow{T_j} A \qquad \text{STEP 2}$$

Both user i and the verifier A form the concatenation of t_j and T_j, $v_j = t_j;T_j$, and calculate the polyrandom function $f(v_j)$ of the resulting string

$$z_j = f(v_j).$$

Since v_j is the joint result of contributions by user i and A, it is indeterminate to both, hence no additional redundant information is needed to insure that z_j will also be indeterminate to both of them.

Both i and A now know z_j (a residue mod n_i) which may or may not be a quadratic residue with respect to n_i. Using the by now familiar procedure to associate a unique quadratic residue with z_j, user i calculates a square root, r_j, and sends

$$i \xrightarrow{r_j:(B_2B_{-1})} A$$ STEP 3

Note that z_j is being used effectively as a one-time key, indeterminate to both i and A because of the polyrandom nature of f, to permit user i to give to A an encrypted function of v_j in a form that will allow A to satisfy himself that whoever he is in communication with had to know the factors of n_i. This exchange does not provide any information about the factors themselves because of the polyrandom nature of f.

If the person seeking to be recognized as user i really is who he claims to be, i.e., if he knows p_i and q_i, then

(6) $$r_j^2 = (2)^{B_2}(-1)^{B_{-1}}(z_j) \qquad (\text{mod } n_i)$$

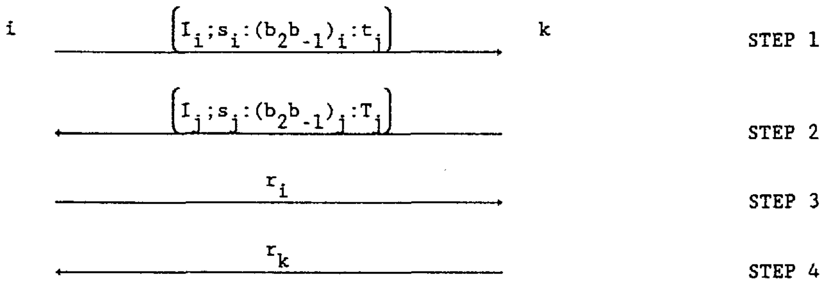
will be satisfied. However, if he is not user i, so that he doesn't know the factorization of n_i, then in order for him to be able to impersonate i, he must find a number x such that

(7) $$(2)^{B_2}(-1)^{B_{-1}}(z_j) - x^2 = 0 \qquad (\text{mod } n_i)$$

which is computationally as difficult as factoring n_i. A knows the identity claimed by the applicant from I_i, which he accepts as the proven identity of the applicant if and only if equality (5) is satisfied:

A keeps the 4-tuple $\left[(I_i;s_i):(v_j;r_j)\right]$ as his certified receipt for the transaction. Anyone can later verify all aspects of the transaction: first by validating the credential $(I_i;s_i)$ in exactly the same way that A did using the public part of the issuer's authentication channel, and then by validating the receipt (v_j,r_j) using the public part of user i's authentication channel. This proves, in probability, that the complete description of the transaction, v_j, was endorsed by user i, or at least by someone knowing the factorization of n_i. As has already been mentioned, the missing B_2B_{-1} and $(b_2b_{-1})_i$ can be (effectively) calculated when needed, and since the frequency of arbitration is expected to be very low compared with the frequency of authentication and retention of receipts which must occur for every transaction, it is more efficient to not store the bits indicating which of the four test residues should be a quadratic residue.

If both communicants require a certified receipt the one-way protocol described above can be easily modified into a two-way protocol between two parties, i and k, both of whom must possess identification credentials validated by the issuer. The exchange in this case is of the form


i
$\left[I_i;s_i:(b_2b_{-1})_i:t_j\right]$
k
STEP 1
$\left[I_j;s_j:(b_2b_{-1})_j:T_j\right]$
STEP 2
r_i
STEP 3
r_k
STEP 4

where user i would keep the 4-tuple $\left[(I_j,s_j):(v_j,r_k)\right]$ as his certified receipt, etc.

We will next prove that the protocol just described is secure. As a matter of fact, we will prove rather substantially more. A number of authors [3,17,18] have devised schemes for embedding a subliminal channel into digital signature or identification schemes. Consequently, for some applications (such as treaty verification) where a subliminal channel could be exploited by one of the parties to cheat the other, it may be essential for a scheme to be acceptable that a means be available to prove that no subliminal channel has been concealed. In [4] van de Graaf and Peralta present a scheme for proving that a modulus n is a Blum integer, and this provides some protection against subliminal channels in identification schemes using Blum integers. We present a zero-knowledge scheme for proving that a modulus n is of the form used here. This will eliminate the possibility of those subliminal channels arising from the modulus n being of either of the forms $n = p^2q$, r $n = pqr$ or $n = p^2pqr$. A great advantage of the identification scheme described here over schemes based on Blum integers is the avoidance of computing Jacobi symbols. Our proof that a modulus n is of the correct form also avoids computing Jacobi symbols.

Since one of the authors is from Texas where the effete Alice and Bob of cryptology fame haven't gained acceptance, and the other is an engineer accustomed to using the notation Tx and Rx to indicate the transmitter and receiver, respectively, in a communications channel, the communicants here will be called Tex and Rex (pronounced with a nasal Texas drawl). With this explanation of the change in notation, we start by assuming that Tex wishes to establish his identity to Rex. A simplified description of the protocol described above is:

1) Tex chooses a string of symbols x and sends it to Rex.
2) After receiving x, Rex chooses a string y and sends it to Tex.
3) They compute $z = f(v)$, where f is a polyrandom function, and $v = x;y$ is the concatenation of the strings x and y.
4) Tex determines which one of the four numbers z, -z, 2z, -2z is a square. Let's say that uz is a square. Then Tex calculates and chooses at random one out of the four possible square roots of uz, say s. He gives s to Rex along with a two-bit suffix (b_2b_{-1}) indicating which of the four numbers

1, 2, -1, or -2 must be used as a multiplier for u to make the product be a square.

5) Rex accepts the communication as authentic if and only if the equality

$$s^2 \equiv (2)^{b_2}(-1)^{b_{-1}}z$$

is satisfied.

As pointed out earlier, there is a potentially troubling aspect to this scheme: Every time that Tex uses it, Rex might conceivably learn something about $n = pq$. If Tex identifies himself k times to Rex, or if k different people to whom Tex has identified himself pool their knowledge, then Rex obtains 2k bits of information about p and q which -- we might naively assume -- have required 2^{2k} guesses in order for him to simulate for himself. That is, if we postulate that he had a procedure for factoring the modulus which required these numbers, and he didn't have them, then he would have had to run his algorithm 4^k times, once for each guess. Instead the algorithm is a zero-knowledge proof, and contrary to intuition, Rex can, on his own, come up with number triples (z,s,u), where z is random, u is in the set $S = \{1,-1,2,-2\}$, and $s^2 = uz$. In other words, we show that he gains no information by Tex's responses that he couldn't get for himself. Acting purely on his own, with no participation by Tex, Rex carries out the following sequence of steps.

1) Pick a random s,

2) pick u randomly in S, and

3) define z by $z = u^{-1}s^2$ (mod n).

These steps can be carried out without knowing the factorization of the modulus n.

Rex can form as many such triples (z,s,u) as he wishes, and they come from the same probability distribution as the ones he obtains from Tex. Hence they don't add to his knowledge, and the protocol is a zero-knowledge proof. We required that the square root s be chosen at random from among the four possible square roots of uz. This is necessary in order that the zero-knowledge argument will hold. It does have the one annoying feature that we must arrange that the probability that Tex chooses the same x twice be negligibly small, since a repetition of z would enable Rex to factor the modulus with probability 1/2.

We next prove that the protocol permits a zero-knowledge proof that the modulus n is of the form $n = pq$, $p \equiv 3$ (mod 8) and $n \equiv 7$ (mod 8), as claimed. This proof process requires two steps. The first protocol proves that n is square-free by demonstrating Tex's ability to take n-th roots. Simmons [18] has embedded a subliminal channel into a digital signature scheme devised by Brickell and DeLaurentis [1] using a modulus of the form $n = p^2q$, which shows that even a modulus with only two distinct prime factors can be a problem.

The second protocol then establishes that the modulus n is indeed of the claimed form: $n = pq$. This is needed, of course, to eliminate the first known subliminal channel (due also to Simmons [17]) which requires a modulus that is the product of three primes: either $n = pqr$ or $n = p^2qr$. At the same time, a new subliminal channel based on $n = pq$, where p and q are not of the right form, is eliminated also.

Protocol for proving n square free.

1) Tex chooses x and sends it to Rex.
2) After receiving x, Rex chooses y and sends it to Tex.
3) They both compute $z = f(v)$, where $v = x;y$ is the concatenation of x and y.
4) Tex finds the n-th root s of z, and sends s to Rex.
5) They repeat steps 1-4 a total of k times.

The basic observation, as explained in [2], is that if n is square free, then every number will have an n-th root, whereas if n is divisible by p^2, where p is a prime, then at most 1/p of the numbers will have n-th roots. Since n is presumably odd, so that $p \geq 3$, there is a probability of at most 3^{-k} that a modulus which is not square free would survive the protocol.

It is important that Tex sends x to Rex _before_ Rex chooses y, to prevent Tex from using the following forward search [11] technique:

1) Tex receives y from Rex.
2) Tex chooses x at random and computes $z = f(v)$, where $v = x;y$.
3) Tex checks whether z has an n-th root. This will happen with probability 1/p if, e.g., $n = p^2q$.
4) If z has an n-th root s, then Tex sends x and then s to Rex.
5) If z does not have an n-th root, then go to step 2.

We remark that the choice of a prime p as small as $p = 3$ is not impossible, since the malefactor may be willing to take risks in order to conceal a subliminal channel. Thus would give Tex's forward search strategy a probability of $1 - 2/3)^k$ of working within k tries. We could, of course, test n for divisibility by primes $3,5,\ldots,p_r$ and reduce this probability to $1 - (1-1/p_r)^k$.

As explained in [13], the protocol doesn't work if the primes are of a special form. For our purposes, $n = pq$, and the protocol will fail if p divides q-1 exactly, or if q divides p-1 exactly. In these cases not all numbers will have n-th roots, and so n would appear to be a bad modulus even though it is not. This is not a serious restriction.

The algorithm gives a zero-knowledge proof, since Rex could produce random pairs (x,z), by choosing z at random and computing $x = z^n \pmod{n}$. These pairs have the same probability distribution as the pairs (x,z) occurring in the protocol.

Protocol for proving n is of the proper form. Using the following protocol, Tex convinces Rex that $n = pq$, where p is a prime $= 3 \pmod{8}$ and q is a prime $= 7 \pmod{8}$:

1) Tex chooses x, Rex chooses y, they compute z = f(x,y).

2) Tex finds the u in {1,-1,2,-2} such that uz is a square, and randomly chooses s, one of the four square roots of uz.

3) Tex sends s and u to Rex.

4) Steps 1 to 3 are repeated k times.

We may assume that the n-th root algorithm has already been applied and hence that n is square-free. If n has three or more prime factors, then at most n/8 of the numbers are squares, and the probability that one of the four numbers z, -z, 2z, -2z is a square is at most 50%. Hence the probability of Tex fooling Rex after k steps is at most 2^{-k}.

How do we know that p = 3 (mod 8) and q = 7 (mod 8)? The answer is that if the modulus isn't of the proper form, that for some choices of a residue u, that no member of the set (u,-u,2u,-2u) will be a square so that Tex can't respond to the challenge value u. For example, p = 1 (mod 8) and q = 3 (mod 8), then 2 is a square mod p and a nonsquare mod q, and -1 is a square mod p and a nonsquare mod q. This means that z will be a square whenever -2z is, os that a 25% probability exists that for any particular z, none of the numbers z, -z, 2z, -2z are squares.

In such a case, the probability that Tex will fool Rex into accepting a modulus which is not of the proper form is at most $(3/4)^k$.

References

1. E. F. Brickell and J. M. DeLaurentis, "An Attack on a Signature Scheme Proposed by Okamoto and Shiraishi," Crypto'85, Santa Barbara, CA, Aug. 19-22, 1985, in Advances in Cryptology, Ed. by H. C. Williams, Springer-Verlag, Berlin, 1986, pp. 28-32.

2. David M. Burton, Elementary Number Theory, Allyn and Bacon, Inc., Boston, MA, 1976.

3. Y. Desmedt, C. Goutier and S. Bengio, "Special Uses and Abuses of the Fiat-Shamir Passport Protocol," preprint obtained from authors.

4. J. van de Graaf and R. Peralta, "A Simple and Secure Way to Show the Validity of your Public Key," Crypto'87, Santa Barbara, CA, Aug. 16-20, 1987, in Advances in Cryptology, Ed. By Carl Pomerance, Springer-Verlag, Berlin, 1988, pp. 128-134.

5. D. E. Knuth, The Art of Computer Programming, Addison-Wesley, Reading, MA, 1969; 2nd ed., 1981.

6. D. H. Lehmer, "Computer Technology Applied to the Theory of Numbers," in MAA Studies in Mathematics, Vol. 6, Studies in Number Theory, W. J. LeVeque, ed., Prentice-Hall, NJ, 1969, pp. 117-151.

7. P. D. Merillat, "Secure Stand-Alone Positive Personnel Identity Verification System (SSA-PPIV)," Sandia National Laboratories Tech. Rpt. SAND79-0070, March.

8. R. M. Needham and M. Schroeder, "Using Encryption for Authentication in Large Networks of Computers," Comm. ACM, Vol. 21(12), Dec. 1978, pp. 993-999.

9. H. Ong, C. P. Schnorr and A. Shamir, "An Efficient Signature Scheme Based on Quadratic Equations," in Proc. 16th Symp. on the Theory of Computing, Washington, 1984, pp. 208-216.

10. H. Ong. C. P. Schnorr and A. Shamir, "Efficient Signature Schemes Based on Polynomial Equations," in Proc. Advances in Cryptology -- Crypto'84 (G. R. Blakley and D. Chaum, Eds.), Lecture Notes in Computer Science 196. New York: Springer-Verlag, 1985, pp. 37-46.

11. J. M. Pollard and C. P. Schnorr, "An Efficient Solution of the Congruence $x^2 + ky^2 = m(\bmod n)$," IEEE Trans. Info. Theory, V. IT-33, No. 5, Sept. 1987, pp. 702-709.

12. G. P. Purdy, "A High Security Log-in Procedure," Comm. ACM, Vol. 17(8), Aug. 1974, pp. 442-445.

13. G. P. Purdy, "A Zero-Knowledge Proof Scheme Showing that $n = pq$," preprint.

14. M. O. Rabin, "Digitized Signatures and Public-key Functions as Intractable as Factorization," M.I.T. Lab. for Computer Science, Tech. Report LCS/TR-212, 1979.

15. G. J. Simmons and D. B. Holdridge, "Forward Search as a Cryptanalytic Tool Against a Public Key Privacy Channel," Proc. of the IEEE Computer Soc. 1982 Symp. on Security and Privacy, Oakland, CA, April 26-28, 1982, pp. 117-128.

16. G. J. Simmons, "A System for Verifying User Identity and Authorization at the Point-of-Sale or Access," Cryptologia, Vol. 8(1), Jan. 1984, pp. 1-21.

17. G. J. Simmons, "The Subliminal Channel and Digital Signatures," Eurocrypt'84, Paris, France, April 9-11, 1984, in Advances in Cryptology, Ed. by T. Beth, et al., Springer-Verlag, Berlin, 1985, pp. 364-378.

18. G. J. Simmons, "A Secure Subliminal Channel (?)," Crypto'85, Santa Barbara, CA, Aug. 19-22, 1985, in Advances in Cryptology, Ed. by H. C. Williams, Springer-Verlag, Berlin, 1986, pp. 33-41.

19. G. J. Simmons, "An Impersonation-Proof Identity Verification Scheme," Proceedings of Crypto'87, Santa Barbara, CA, August 16-20, 1987, in Advances in Cryptology, Ed. by Carl Pomerance, Springer-Verlag, Berlin, to appear.

20. J. Stein, "Computational Problems Associated with Racah Algebra," J. Comp. Phys., Vol. 1, 1967, pp. 397-405.

21. M. V. Wilkes, Time-Sharing Computing Systems, Elsevier/MacDonald, New York, 1968; 3rd ed., 1975.

22. H. C. Williams, "A Modification of the RSA Public-Key Encryption Procedure," IEEE Trans. on Info. Theory, Vol. IT-26, No. 6, Nov. 1980, pp. 726-729.