Remote scheme for password authentication based on theory of quadratic residues

Chin-Chen Chang*, Sun-Min Tsu* and Chien-Yuan Chen†

We propose a remote password authentication scheme based on quadratic residues. In our scheme, any legal user can freely choose his own password in the card initialization phase. Using his password and smart card which contains identity and other information, he can then log into the system successfully. According to our analysis, intruders cannot obtain any secret information from the public information, or derive any password from intercepted messages. In addition, our scheme can withstand the attack of replaying previously intercepted log-in requests.

Keywords: remote password authentication, quadratic residues, login request, time stamp, smart card

With the rapid development of science and technology, people rely increasingly on networks to communicate with others and to have their jobs run on remote hosts. Computer networks provide convenient procedures for users operating at remote places. An eavesdropper can easily access and intercept the information transmitted. Thus, the need to provide protection and security arises, especially when a user operates from a remote terminal.

The most conventional way to achieve password authentication is by means of password tables in the host machine, which contain identities (ID) and their corresponding passwords (PW) for each legal user. One direct way to authenticate passwords is to maintain a password table for further verification. However, this approach cannot avoid the threat of revealing passwords.

Approaches[1–6] have been proposed to eliminate the problem by encoding plain passwords as test patterns or verification patterns instead of having plain password tables in the system. These schemes cannot withstand the attack of replaying previously intercepted log-in information.

In a remote access system, an eavesdropper can impersonate the legal user to log-in to the system in a later log-in by intercepting the legal log-in information. Lamport[7] proposed a scheme to protect against such an attack, but it is insecure if the encrypted passwords in the centre are modified by a malicious intruder. Harn[8] proposed a concept using a dynamic password to prevent this attack, but his scheme needs a table to store the ID and the log-in time, thus making the system more insecure.

Chang and Wu[9] proposed a scheme with a smart card for remote password authentication to overcome such a threat. However, their scheme was shown to be insecure and breakable by Chang and Laih[10]. Chang and Hwang[11] also presented another remote password authentication scheme using a smart card. The security of their scheme is based upon the discrete logarithm problem, but in these schemes the password of the user must be assigned by the system. This assumption is not reasonable. Our scheme, therefore, is motivated by getting rid of this assumption.

This paper is organized as follows. In the next section, we review briefly Chang and Hwang's remote password authentication scheme. We then introduce Harn and Kiesler's probabilistic encryption scheme, because we use it to develop a new remote password authentication. Our remote scheme for password authentication is described, and some security analyses are presented. Finally, some conclusions are given.

CHANG AND HWANG'S REMOTE PASSWORD AUTHENTICATION SCHEME

Let P be a large prime and σ be a primitive element of the Galois field $GF(P)$. The system must select a secret

*Institute of Computer Science and Information Engineering, National Chung Cheng University, Chiayi, Taiwan 621, ROC
†Institute of Computer and Information Science, National Chiao Tung University, Hsinchu, Taiwan 300, ROC
Paper received: 2 September 1994

0140-3664/95/$09.50 © 1995–Elsevier Science B.V. All rights reserved

nonsingular key matrix **K** with $n \times n$ dimension as follows:

$$\mathbf{K} = \begin{bmatrix} k_{11}, & k_{12}, \dots, & k_{1n} \\ \vdots & \vdots & \vdots \\ k_{n1}, & k_{n2}, \dots, & k_{nn} \end{bmatrix},$$

where n is a positive integer and k_{ij} belongs to $GF(P)$ for $i, j = 1, 2, \dots, n$. In addition, the system publishes a one-way function $f(\cdot)$ and chooses a pseudorandom number generator function $g(\cdot)$ which has to be kept secret. Next, the system computes a public key matrix:

$$\mathbf{PK} = \begin{bmatrix} pk_{11}, & pk_{12}, \dots, & pk_{1n} \\ \vdots & \vdots & \vdots \\ pk_{n1}, & pk_{n2}, \dots, & pk_{nn} \end{bmatrix},$$

where $pk_{ij} = \sigma^{k_{ij}} \bmod P$ for $i, j = 1, 2, \dots, n$.

In general, there are three phases in remote password authentication: *card initialization*, *login* and *authentication*.

Card initialization phase

The system assumes that there is a trusted Card Initialization Centre (CIC). The centre executes the card-issuing operation when a new user U_i registers to the system. Then, user U_i submits his identity ID_i to the CIC; the CIC then uses the following steps to issue a smart card containing a key matrix **PK** and a one-way function $f(\cdot)$.

Card initialization procedure

Input: user U_i's identity number $ID_i = (id_{i1}, id_{i2}, \dots, id_{in})$ and the system secret key **K**.

Output: the password $PW_i = (pw_{i1}, pw_{i2}, \dots, pw_{in})$.

Step 1: Compute

$$PW_i = \begin{bmatrix} pw_{i1} \\ pw_{i2} \\ \vdots \\ pw_{in} \end{bmatrix} = \begin{bmatrix} k_{11}, & k_{12}, \dots, & k_{1n} \\ \vdots & \vdots & \vdots \\ k_{n1}, & k_{n2}, \dots, & k_{nn} \end{bmatrix}^{-1} \begin{bmatrix} g(id_{i1}) \\ g(id_{i2} \\ \vdots \\ g(id_{in}) \end{bmatrix},$$

where $\mathbf{K}^{-1}$ is the inverse of matrix **K**.

Step 2: Deliver a smart card containing a key matrix **PK** and a one-way function $f(\cdot)$ to the user U_i. But the password PW_i must be sent to user U_i through a secret channel.

Login phase

When a user U_i wants to log-in to the system remotely, he has to attach his smart card to a remote terminal and input his password PW_i. The login request is constructed as follows.

Login procedure

Input: the password PW_i, and the current log-in time T.

Output: The login request R.

Step 1: Randomly choose an integer vector $\mathbf{V} = (v_1, v_2, \dots, v_n)$, where $v_i \in GF(P)$.

Step 2: Compute a vector $\mathbf{S} = (s_1, s_2, \dots, s_n)$, where:

$$s_m = \prod_{j=1}^{n} pk_{mj}^{\ v_j} \bmod P \quad \text{for } m = 1, 2, \dots, n$$

Step 3: Compute:

$$PW' = (pw'_1, pw'_2, \dots, pw'_n) = PW_i + (V * f(S, T) \bmod (P-1))$$

Step 4: [Construct the login request R]: Construct $R = (ID_i, PW', S, T)$ as the login request.

Authentication phase

Assume that the system receives the log-in request message $R = (ID_i, PW', S, T)$ from user U_i at time T^*; then the system uses the following steps to verify the request:

Authentication procedure

Input: The login request message $R = (ID_i, PW', S, T)$, the message received in time T^*, and the system secret key K.

Output: Accept or Reject the log-in request.

Step 1: Verify whether the format of ID_i is correct. If it is not then the system rejects the log-in request.

Step 2: Verify whether the transmission time (i.e. $T^* - T$) is within the legal tolerant interval ΔT. If $(T^* - T) > \Delta T$, then the request is rejected.

Step 3: Compute:

$$Q = \begin{bmatrix} q_1 \\ q_2 \\ \vdots \\ q_n \end{bmatrix} = \begin{bmatrix} k_{11}, & k_{12}, \dots, & k_{1n} \\ \vdots & \vdots & \vdots \\ k_{n1}, & k_{n2}, \dots, & k_{nn} \end{bmatrix}^{-1} \begin{bmatrix} pw'_1 \\ pw'_2 \\ \vdots \\ pw'_n \end{bmatrix}.$$

Step 4: [Accept or Reject]: If $\sigma^{q_j} \equiv \sigma^{g(id_{ij})} * s_j^{f(S,T)} \pmod P$ for $j = 1, 2, \dots, n$, then accept the log-in request; otherwise reject it.

HARN–KIESLER'S PROBABILISTIC ENCRYPTION SCHEME

Before describing Harn–Kiesler's[14] scheme, we need to define some symbols and some properties of quadratic residues.

Definition 1[12] *If n is a positive integer, we state that the integer a is a quadratic residue modulo n if* $GCD(a, n) = 1$ *and if the congruence* $x^2 \equiv a \pmod n$ *has a solution; if the congruence* $x^2 \equiv a \pmod n$ *has no solution, we state that a is a quadratic non-residue modulo n. Here* $GCD(a, b)$ *means the greatest common divisor of a and b.*

We here use the symbol QR_n to denote the set of all integers in the range from 1 to $n-1$ that are quadratic residues modulo n, and the symbol QNR_n to denote those integers that are quadratic non-residues modulo n.

Definition 2[12] *Let p be an odd prime and a be an integer not divisible by p. The Legendre symbol* $L\left(\frac{a}{p}\right)$ *is defined by:*

$$L\left(\frac{a}{p}\right) = \begin{cases} 1, & \text{if } a \in QR_p \\ -1, & \text{if } a \in QNR_p \end{cases}$$

Definition 3

1. *Bitlength(m) denotes the length of the binary form of the integer m.*
2. *(a, b) denotes the set of the integer i such that $a < i < b$.*
3. *$CRT(n, p, q, \alpha, \beta)$ denotes the integer k, which can be computed by the Chinese Remainder Theorem, such that:*

$$k \equiv \alpha (\mathrm{mod}\, p)$$

$$k \equiv \beta (\mathrm{mod}\, q), \text{ where } n = pq$$

According to Lamport[7], there are four square roots of congruence $x^2 \equiv a (\mathrm{mod}\, n)$. We briefly review this in what follows.

Theorem 4 *For the integer $n = pq$, with p and q being distinct odd primes of form $4k + 3$, and an integer $a \in QR_n$, four square roots of a are distinguishable among four cases stated as:*

1. $\mathrm{root}_1 \in QR_p \cap QR_q$
2. $\mathrm{root}_2 \in QNR_p \cap QR_q$
3. $\mathrm{root}_3 \in QR_p \cap QNR_q$
4. $\mathrm{root}_4 \in QNR_p \cap QNR_q$

In Harn–Kiesler's scheme, each user U_i needs to choose two large distinct prime numbers p_i and q_i, both of form $4k + 3$, and calculate $n_i = p_i \times q_i$. An integer μ_i between 1 and $(n_i - 1)$ is necessary for each user U_i, with $\mu_i \in QNR_{pi} \cap QNR_{qi}$. Then (n_i, μ_i) becomes user U_i's public key; (p_i, q_i) is used as his secret key.

Encryption algorithm

Input: a message $m = m_1 m_2 \ldots m_t$, with $m_i \in \{0, 1\}$ for $1 \leqslant i \leqslant t$, a random number x between 1 and $n_i - 1$, such that $GCD(x, n_i) = 1$, and user U_i's public key (n_i, μ_i).

Output: Ciphertext C.

Initial: $r = 1$, and $t = \mathrm{BitLength}(m)$.

Step 1: [Calculate C_r]

$$C_r = \begin{cases} x^2 \bmod n_i, & \text{if } m_r = 0 \\ x^2 \times \mu_i \bmod n_i, & \text{if } m_r = 1 \end{cases}$$

Step 2 [Set indication bit b_r]:

$$b_r = \begin{cases} 0, & \text{if } C_r \in \left(0, \frac{n_i}{2}\right) \\ 1, & \text{if } C_r \in \left(\frac{n_i}{2}, n_i\right) \end{cases}$$

Step 3 [Calculate C_{r+1} from C_r]:

$$C_{r+1} = \begin{cases} (C_r)^2 \bmod n_i, & \text{if } m_{i+1} = 0 \\ (Cr)^2 \mu_i \bmod n_i, & \text{if } m_{i+1} = 1 \end{cases}$$

Step 4: Compute $r = r + 1$. If $r \leqslant t$ then go to Step 2; otherwise continue.

Step 5 [Construct ciphertext C]:

$$C = (C_t, b_{t-1} \| b_{t-2} \| \ldots b_1)$$

in which '$\|$' is the concatenation operator.

Then the ciphertext C is sent to user U_i. When user U_i receives ciphertext C from user U_j, he can recover the plaintext by using the decryption algorithm.

Decryption algorithm

Input: Ciphertext $C = (C_t, b_{t-1} \| b_{t-2} \| \ldots \| b_1)$, and secret key (p_i, q_i).

Output: Plaintext $m = m_1 m_2 \ldots m_t$.

Initial: $r = 1$, and $t = \mathrm{BitLength}(b_{t-1} b_{t-2} \ldots b_1) + 1$.

Step 1: $\left[\text{Calculate } L\left(\frac{C_{t-r+1}}{p_i}\right) \text{ and } L\left(\frac{C_{t-r+1}}{q_i}\right)\right]$

$$\text{Compute } L\left(\frac{C_{t-r+1}}{p_i}\right) = (C_{t-r+1})^{(p_i - 1)/2} \bmod p_i$$

$$\text{Compute } L\left(\frac{C_{t-r+1}}{q_i}\right) = (C_{t-r+1})^{(q_i - 1)/2} \bmod q_i$$

Step 2 [Determine the message bit m_{t-r+1}]:

$$m_{t-r+1} = \begin{cases} 0, & \text{if } C_{t-r+1} \in QR_{p_i} \cap QR_{q_i} \\ 1, & \text{if } C_{t-r+1} \in QNR_{p_i} \cap QNR_{q_i} \end{cases}$$

Step 3 [Specify a congruence equation]:

$$C_{t-r} = \begin{cases} \text{a specified root of } x^2 \equiv C_{t-r+1} (\mathrm{mod}\, n_i) & \text{if } m_{t-r+1} = 0 \\ \text{a specified root of } x^2 \equiv C_{t-r+1} (\mu_i)^{-1} (\mathrm{mod}\, n_i) & \text{if } m_{t-r+1} = 1 \end{cases}$$

Step 4 [Find a possible specified solution C_{t-r}] Construct a pair (α, β), with $\alpha \in QR_{p_i}$ and $\beta \in QR_{q_i}$, from the congruence equation that is determined at Step 3. Compute $C_{t-r} = CRT(n_i, p_i, q_i, \alpha, \beta)$.

Step 5 [Determine the specified solution C_{t-r}]:

Case 1: $b_{t-r} = 0$.

If $C_{t-r} \in \left(0, \frac{n_i}{2}\right)$ then return; otherwise

$C_{t-r} = n - C_{t-r}$.

Case 2: $b_{t-r} = 1$.

If $C_{t-r} \in \left(\frac{n_i}{2}, n_i\right)$ then return; otherwise

$C_{t-r} = n - C_{t-r}$.

Step 6: $r = r + 1$. If $r \leqslant t$ then goto Step 1; otherwise continue.

Step 7 [Construct the plaintext message m]: $m = m_1 \| m_2 \| \ldots \| m_t$. Finally, message bit m_r is found for $1 \leqslant r \leqslant t$; the original plaintext m is recovered.

Example 5

Assume that user U_j wants to send a message $m = 437 = (110110101)_2$ to user U_i. Suppose that user U_i's secret key $(p_i, q_i) = (11, 19)$, and his public key $(n_i, \mu_i) = (209, 41)$.

Encryption

Input: $m = 437 = (110110101)_2 = m_1 m_2 \ldots m_9$. User U_i's public key $(n_i, \mu_i) = (209, 41)$, and a random number $x = 75$, where $(75, 209) = 1$.

Output: the ciphertext C.
Initial: $r = 1$ and $t = \text{BitLength}(437) = 9$.

Step 1 [Calculate C_r]: Since $m_r = 1$, compute
$C_r = x^2 \times \mu_i \bmod n_i$
$= 75^2 \times 41 \bmod 20$
$= 98$

Step 2 [Set the indication bit b_r]: $b_r = 0$ because $C_r \in \left(0, \frac{209}{2}\right)$.

Step 3: As $m_{r+1} = 1$, compute $C_{r+1} = (98)^2 \times 41 \bmod 209 = 8$.

Step 4: Compute $r = 1 + 1 = 2$; go to Step 2.

Repeating the enciphering process eight times, we obtain the results shown in *Table 1*.

Step 5: The ciphertext C is equal to $C = (129, (10011000)_2 = (129, 152)$.

Decryption
Input: Ciphertext $C = (129, 152)$, secret key $(p, q) = (11, 19)$.
Output: Plaintext $m = m_1 m_2 \ldots m_t$.
Initial: $r = 1$, and $t = \text{BitLength}(152) + 1 = 9$.

Step 1: Compute $L\left(\frac{129}{11}\right) = -1$.

Compute $L\left(\frac{129}{9}\right) = -1$.

Step 2: The message bit $m_9 = 1$ since $129 \in QNR_{11} \cap QNR_{19}$.

Step 3: C_8 is a specified square root of

$$x^2 \equiv 129 \times 41^{-1} \pmod{209}$$
$$\equiv 129 \times 51 \pmod{209}$$
$$\equiv 100 \pmod{209}.$$

Step 4: [Construct a pair (α, β), where $\alpha \in QR_{11}$ and $\beta \in QR_{19}$]
Compute $x^2 = 100 \bmod 11 = 1$.
Compute $x^2 = 100 \bmod 19 = 5$.
Compute $\alpha = 1^{(11+1)/4} \bmod 11 = 1 \in QR_{11}$.
Compute $\beta = 5^{(19+1)/4} \bmod 19 = 9 \in QR_{19}$.
Compute $C_8 = CRT(209, 11, 19, 1, 9) = 199$.

Step 5: Since the indication bit $b_8 = 1$, $C_8 = 199$ is exactly determined.

Repeating from Steps 1 to 5 eight times, then $m_8 = 0$, $m_7 = 1$, $m_6 = 0$, $m_5 = 1$, $m_4 = 1$, $m_3 = 0$, $m_2 = 1$ and $m_1 = 1$ are recovered. The plaintext $m = (m_1 \| m_2 \| \ldots \| m_9)_2 = (110110101)_2 = 437$ is recovered back.

Table 1 Results in the encryption process.

r	1	2	3	4	5	6	7	8	9
m_r	1	1	0	1	1	0	1	0	1
C_r	98	8	64	109	151	20	98	199	129
b_r	0	0	0	1	1	0	0	1	×

REMOTE SCHEME FOR PASSWORD AUTHENTICATION

There are three phases of our proposed implementation of remote password authentication: *card initialization*, *log-in* and *authentication*.

Card initialization phase

The system assumes that there is a trusted card initialization centre (CIC). The centre executes the card-issuing operation when a new user U_i registers to the system. The system keeps two large odd primes, p and q, as the secret key pair (p, q), and uses the pair (n, μ) as the public key, with $n = pq$ and $\mu \in QNR_p \cap QNR_q$. Besides, the system also publishes a one-way hashing function $f(\cdot)$. Initially, a new user U_i freely selects a password w_i to compute $f(w_i, 0)$. Then, user U_i submits his identity ID_i and $f(w_i, 0)$ to the CIC; then the CIC uses the following steps to issue a smart card containing an extended password PW_i and an account AC_i for the new user U_i:

Card initialization procedure

Input: user U_i's identity number ID_i, $f(w_i, 0)$, and the system public key pair (n, μ).
Output: the account number AC_i and the extended password PW_i.

Step 1 [Choose a random number x]: Randomly choose a number x with $GCD(x, n) = 1$.

Step 2 [Evaluate the binary form of the ID_i]: Let $ID_i = (m_1 m_2 \ldots m_t)_2$, where $m_i \in \{0, 1\}$ for $1 \leqslant i \leqslant t$, and t is the bit length of the ID_i.

Step 3 [Calculate the cryptogram C_i and the binary bit b_i]: Initially, set $C_0 = x$, and for i from 1 to t do:
If $m_i = 0$ then $\{C_i = (C_{i-1})^2 \bmod n\}$; else $\{C_i = (C_{i-1})^2 \times \mu \bmod n\}$
If $C_i \in \left(0, \frac{n}{2}\right)$ then set bit $b_i = 0$; else set bit $b_i = 1$.

Step 4 [Generate account AC_i and extended password PW_i]: Use C_t as the account number AC_i and view $(b_{t-1} b_{t-2} \ldots b_1)_2$ as W_i. Then, compute the extended password $PW_i = W_i \oplus f(w_i, 0)$. Here $a \oplus b$ denotes *XOR* of the binary forms of a and b.

Step 5 [Test whether the account number AC_i has existed in the system]: If account AC_i has existed in the account file, then go to Step 1; otherwise continue.

Step 6 [Issue a smart card]: A smart card containing the $(ID_i, AC_i, PW_i, f(\cdot), n, \mu)$ is issued to the new user U_i, in which $f(\cdot)$ is a one-way hashing function.

Example 6

Assume that a new user Bob, with $ID = 437$ and $f(w, 0) = 234$, where $w = 43$, seeks to register to the system, then the encryption process is the same case as in Example 5. The CIC issues a smart card to Bob, which contains his $ID = 437$, $AC = 129$, $PW = W \oplus f(w, 0) = 152 \oplus 234 = (10011000)_2 \oplus (11101010)_2 = (01110010)_2 = 104$, system public key $n = 209$, $\mu = 41$, and a public one-way function f.

Log-in phase

When a user U_i wants to log-in to the system remotely, he has to attach his smart card to a remote terminal and input the real password w_i. The log-in request is constructed as follows.

Login Procedure
Input: Real password w_i, and the current login time T.
Output: The login request R.

Step 1: Randomly choose an integer x within the interval $[1, n-1]$, with $GCD(x,n) = 1$.

Step 2 [Determine the indication information (d_1, d_2)]:

$$d_1 = \begin{cases} 0 & \text{if } x \text{ is odd} \\ 1 & \text{if } x \text{ is even} \end{cases} \quad \text{and} \quad d_2 = \begin{cases} 0 & \text{if } x \in \left(0, \frac{n}{2}\right) \\ 1 & \text{if } x \in \left(\frac{n}{2}, n\right) \end{cases}$$

Step 3 [Calculate the indirect password PW_{i1}]. Compute $W_i = PW_i \oplus f(w_i, 0)$ and $PW_{i1} = W_i \times f(x, T) \bmod n$.

Step 4 [Generate the media ciphertext C_{pw}]: Let $m_1 m_2 \ldots m_\delta$ be the binary form of PW_{i1}, in which δ is the bit length of PW_{i1}. Initially, set $C_0 = x$, for i from 1 to δ do

If $m_i = 0$ then $C_i = (C_{i-1})^2 \bmod n$; else $C_i = (C_{i-1})^2 \times \mu \bmod n$.

If $C_i \in \left(0, \frac{n}{2}\right)$ then $b_i = 0$; else $b_i = 1$.

Let $C_{pw} = {C_\delta}^2 \bmod n$ and $B = b_\delta \| b_{\delta-1} \| b_{\delta-2} \| \cdots \| b_1$.

Step 5 [Generate the pseudo password PW_{i2}]. Construct $PW_{i2} = (B \| d_1 \| d_2) = (b_\delta \| b_{\delta-1} \| b_{\delta-2} \| \cdots \| b_1 \| d_1 \| d_2)$.

Step 6 [Construct the login request R]: Construct $R = (ID_i, AC_i, C_{pw}, PW_{i2}, T)$ as the login request.

Example 7

Bob brings his password $w = 43$ and seeks to log-in to the system remotely at time $T = 1993/08/02/12{:}32$, and his smart card contains his $ID_{\text{Bob}} = 437$, $AC_{\text{Bob}} = 129$, $PW_{\text{Bob}} = 104$, system public keys $n = 209$, $\mu = 41$, and a public one-way function $f(\cdot)$. The log-in request is computed according to the following steps:

Step 1: Randomly choose an integer $x = 139$, with $(139, 209) = 1$.

Step 2: Set $(d_1, d_2) = (0, 1)$.

Step 3: Assume that $f(139, 1993/08/02/12{:}32 = 96$ and $f(43, 0) = 234$; compute:

$$W_{\text{Bob}} = 234 \oplus 104 = 152 \quad \text{and}$$
$$PW_{\text{Bob1}} = 152 \times 96 \bmod 209 = 171$$

Step 4: As $171 = (10101011)_2$, compute C_{pw} and B as follows:

$C_1 = 139^2 \times 41 \bmod 209 = 51, \quad b_1 = 0$
$C_2 = 51^2 \bmod 209 = 93, \quad b_2 = 0$
$C_3 = 93^2 \bmod 209 = 145, \quad b_3 = 1$
$C_4 = 145^2 \bmod 209 = 125, \quad b_4 = 1$
$C_5 = 125^2 \times 41 \bmod 209 = 40, \quad b_5 = 0$
$C_6 = 40^2 \bmod 209 = 137, \quad b_6 = 1$
$C_7 = 137^2 \times 41 \bmod 209 = 200, \quad b_7 = 1$
$C_8 = 200^2 \times 41 \bmod 209 = 186, \quad b_8 = 1$

Compute $C_{pw} = 186^2 \bmod 209 = 111$.
Compute $B = (11101100)_2 = 472$.

Step 5 [Construct the pseudo password PW_{Bob2}]: Compute $PW_{\text{Bob2}} = (B \| d_1 \| d_2) = (1110110001)_2 = 945$.

Step 6 [Construct the login request R]:

$$R = (ID_{\text{Bob}}, AC_{\text{Bob}}, C_{\text{pw}}, PW_{\text{Bob}}, T) = (437, 129, 111, 945, 1993/08/02/12{:}32).$$

Authentication phase

Assume that the system receives the log-in request message $R = (ID_i, AC_i, C_{pw}, PW_{i2}, T)$ from user U_i at time T^*; then the system uses the following steps to verify the request.

Authentication procedure

Input: The login request message $R = (ID_i, AC_i, C_{pw}, PW_{i2}, T)$, the message received time T^*, and the system secret key pair (p, q).
Output: Accept or Reject the login request.
Initial: $r = 1$, $t = \text{BitLength}(PW_{i2}) - 2$, and $PW_{i2} = (b_{t+2} b_{t+1} \ldots b_1)_2$.

Step 1 Verify whether the format of ID_i is correct. If it is not then the system rejects the log-in request.

Step 2: Verify whether the transmission time (i.e. $T^* - T$) is within the legal tolerant interval ΔT. If $(T^* - T) > \Delta T$, then the request is rejected.

Step 3 [Find a possible specified solution for C_{t-r+1}]:
Compute $\alpha = (C_{pw} \bmod p)^{(p+1)/4} \bmod p$
Compute $\beta = (C_{pw} \bmod q)^{(q+1)/4} \bmod q$
If $\alpha \in QNR_p$ then compute $\alpha = p - \alpha$
If $\beta \in QNR_q$ then compute $\beta = q - \beta$
Compute $C_{t-r+1} = CRT(n, p, q, \alpha, \beta)$

Step 4 [Determine the specified solution C_{t-r+1}]:

Case 1: $b_{t-r+3} = 0$. If $C_{t-r+1} \in \left(0, \frac{n}{2}\right)$ then continue; else $C_{t-r+1} = n - C_{t-r+1}$.

Case 2: $b_{t-r+3} = 1$.

If $C_{t-r+1} \in \left(\frac{n}{2}, n\right)$ then continue; else $C_{t-r+1} = n - C_{t-r+1}$.

Step 5 [Determine the $(t-r+1)$th message bit $m_{pw(t-r+1)}$ of the pseudo password PW_{i1}]:

$$m_{pw(t-r+1)} = \begin{cases} 0 & \text{if } L\left(\frac{C_{t-r+1}}{p}\right) = 1 \text{ and } L\left(\frac{C_{t-r+1}}{q}\right) = 1 \\ 1 & \text{if } L\left(\frac{C_{t-r+1}}{p}\right) = -1 \text{ and } L\left(\frac{C_{t-r+1}}{q}\right) = -1 \end{cases}$$

Step 6 [Find a quadratic congruence equation]:

$$C_{t-r} = \begin{cases} \text{a specified root of } x^2 = C_{t-r+1} \bmod n & \text{if } m_{pw(t-r+1)} = 0 \\ \text{a specified root of } x^2 = C_{t-r+1} \times \mu^{-1} \bmod n & \text{if } m_{pw(t-r+1)} = 1 \end{cases}$$

Construct a pair (α, β) from the quadratic congruence determined above, with $\alpha \in QR_p$ and $\beta \in QR_q$. Compute $C_{t-r} = CRT(n, p, q, \alpha, \beta)$.

Step 7: $r = r + 1$. If $r \leqslant t$ then go to Step 4; else continue.

Step 8 [Construct the indirect password $[W_{i1}]$: Construct $PW_{i1} = (m_{pw1}\|m_{pw2}\| \dots \|m_{pwt})_2$.

Step 9 [Recover the seed s]:

$$s = \begin{cases} \text{a specified root of } x^2 \equiv C_1 \,(\text{mod } n) & \text{if } m_{pw1} = 0 \\ \text{a specified root of } x^2 \equiv C_1 \times \mu^{-1} \,(\text{mod } n) & \text{if } m_{pw1} = 1 \end{cases}$$

Find four solutions from the congruence equation determined above. The seed s is among these four solutions and is found as follows:

$$s = \begin{cases} \text{odd} & \text{if } b_2 = 0 \\ \text{even} & \text{if } b_2 = 1 \end{cases}$$

$$\text{and} \quad s \in \begin{cases} \left(0, \frac{n}{2}\right) & \text{if } b_1 = 0 \\ \left(\frac{n}{2}, n\right) & \text{if } b_1 = 1 \end{cases}$$

Step 10 [Recover the password]: Compute $W_i - PW_{i1} \times (f(s, T))^{-1} \bmod n$.

Step 11 [Construct the verified value ID']: Let $r = 1$, $t = \text{BitLength}(W_i) + 1$, $W_i = (b_{t-1}b_{t-2}\dots b_1)_2$ and let $C_t = AC_i$; repeat from Step 5 again, the message bit m_t, $m_{t-1}, \dots,$ and m_1 can be recovered. ID' is constructed as follows:

$$ID' = (m_1\|m_2\| \dots \|m_t)_2.$$

Step 12 [Accept or Reject]: If $ID' = ID$ then Accept the log-in request; else Reject it.

Example 8

When Bob sends a remote login request $R = (437, 129, 111, 945, 1993/08/02/12{:}32)$, the system verifies this request in the following steps. Suppose that the format of Bob's ID is correct and that the transmission time to receive the log-in request is within the tolerant interval ΔT.

Initial: $r = 1$, $t = \text{BitLength}(945) - 2 = 8$, and $945 = (1110110001) = (b_{10}b_9\dots b_1)$.

Step 1 [Find a possible specified solution of $x^2 \equiv 111$ (mod 209)]:

Compute $\alpha = (111)^{(11+1)/4} \bmod 11 = 1 \in QR_{11}$.
Compute $\beta = (111)^{(19+1)/4} \bmod 19 = 4 \in QR_{19}$.
Compute $C_{t-r+1} = C_8 = CRT(209, 11, 19, 1, 4) = 23$.

Step 2 [Determine the specified solution]: Since $b_{t-r+3} = b_{10} = 1$, compute $C_8 = n - C_8 = 209 - 23 = 186$.

Step 3: Compute $L\left(\frac{C_{t-r+1}}{p}\right) = L\left(\frac{C_8}{11}\right) = L\left(\frac{186}{11}\right) = -1$.

Compute $L\left(\frac{C_{t-r+1}}{q}\right) = L\left(\frac{C_8}{19}\right) = L\left(\frac{186}{19}\right) = -1$.

So $m_{pw(t-r+1)} = m_{pw8} = 1$.

Step 4 [Find a quadratic congruence equation]: As $m_{pw8} = 1$, $C_{t-r} = C_7$ is a specified solution of the congruence $x^2 = 186 \times 41^{-1} \bmod 209 = 81 \bmod 209$. Construct a pair $(\alpha, \beta) = (9, 9)$ from the congruence equation $x^2 = 81 \bmod 209$. Compute $C_{t-r} = C_7 = CRT(209, 11, 19, 9, 9) = 9$.

Step 5: $r = r + 1 = 2$. If $r \leqslant 8$ then go to Step 2; else continue. Repeating the above steps eight times, we obtain

$m_{pw8} = 1$, $C_7 = 200$,
$m_{pw7} = 1$, $C_6 = 137$,
$m_{pw6} = 0$, $C_5 = 40$,
$m_{pw5} = 1$, $C_4 = 125$,
$m_{pw4} = 0$, $C_3 = 145$,
$m_{pw3} = 1$, $C_2 = 93$,
$m_{pw2} = 0$, $C_1 = 51$,
$m_{pw1} = 1$.

Step 6 [Construct indirect password PW_{i1}]:

$$\begin{aligned} \text{Construct } PW_{i1} &= (m_{pw1}\|m_{pw2}\| \dots \|m_{pw8})_2 \\ &= (10101011)_2 \\ &= 171. \end{aligned}$$

Step 7 [Recover the seed s]: As $m_{pw1} = 1$, the seed s is a specified solution of the congruence:

$$\begin{aligned} x^2 &\equiv C_1 \times \mu^{-1} \,(\text{mod } n) \\ &\equiv 51 \times 41^{-1} \,(\text{mod } 209) \\ &\equiv 93. \end{aligned}$$

The four solutions for the congruence $x^2 = 93\,(\text{mod } 209)$ are 51, 70, 139 and 158. Therefore the seed $s = 139$ because $b_2 = 0$ and $b_1 = 1$.

Step 8 [Recover the password]

$$\begin{aligned} \text{Compute } W_{\text{Bob}} & \\ &= 171 \times (f(139, 1993/08/02/12{:}32))^{-1} \bmod 209 \\ &= 171 \times 96^{-1} \bmod 209 \\ &= 152. \end{aligned}$$

Step 9: Again, let $r = 1$, $t = \text{BitLength}(152) + 1 = 9$, $(b_8b_7\dots b_1)_2 = 152 = (10011000)_2$, and $C_9 = AC_{\text{Bob}} = 129$. Repeat from Steps 3 to 5 nine times; the message bits are recovered as the same case in Example 3.5; we have $m_9 = 1$, $m_8 = 0$, $m_7 = 1$, $m_6 = 0$, $m_5 = 1$, $m_4 = 1$, $m_3 = 0$, $m_2 = 1$, and $m_1 = 1$.

Construct $ID' = (m_1\|m_2\| \dots \|m_9)_2 = (110110101)_2 = 437$.

Step 10: As the computed $ID' = ID_{\text{Bob}} = 437$, the system accepts Bob's log-in request.

SECURITY ANALYSIS OF OUR REMOTE AUTHENTICATION SYSTEM

Because our proposed scheme does not need to maintain or store any table in the system, it is unnecessary to consider the problems of password table maintenance and of the threat of modification made by a malicious intruder.

The security of our scheme is based on the difficulty of finding the solutions of the congruence $x^2 \equiv a\,(\text{mod } n)$, in which n is a product of two large primes. Rabin[13] showed that finding the square roots of the congruence is equivalent to the factorization problem.

If an intruder tries to impersonate the legal user U_i by replaying the previous intercepted log-in message $R = (ID_i, AC_i, C_{pw}, PW_{i2}, T)$ in a later log-in, even if he modifies the time stamp T into T' successfully, and assumes that the random seed s is known, the real password PW_i still cannot be recovered correctly as $f(s, T') \neq f(s, T)$.

The random seed s and the indirect password PW_{i1} are embedded in the log-in request. Nobody can find the seed s and the indirect password PW_{i1} from the media ciphertext C_{pw} and the pseudo password PW_{i2}.

Finally, transmitted pseudo password PW_{i2} is distinct for each log-in, hence it protects against the potential chosen-plaintext attack.

CONCLUSIONS

We have proposed a new mechanism for password authentication based on the theory of quadratic residues. The scheme has the following attractive features:

1. The user can freely choose his password.
2. The system doesn't know the user's real password because the user uses $f(w, 0)$ to register with the CIC.
3. The scheme neither needs to store nor maintain verification tables in the system.
4. The proposed scheme uses the concept of a probabilistic password to enforce the security of the remote access request.
5. The scheme is useful for authentication in the remote access system when remote messages are sent through insecure communication channels.
6. Our scheme is more suitable for the application of smart cards than Chang and Hwang's[2] authentication scheme. It is important to notice that the amount of computing of the smart card should be as small as possible. In Chang and Hwang's scheme, the number of operations made by the smart card needs at least n modular exponentiations. However, our scheme only needs n modular multiplications and modular squaring operations, much less than that needed by Chang and Hwang's scheme.

REFERENCES

1 Chang, C C and Wu, L H 'A password authentication scheme based upon Rabin's public-key cryptosystems', *Proc. Int. Conf. System Management '90*, Hong Kong (June 1990) pp 425–429
2 Evans, A, Jr, Kantrowitz, W and Weiss, E 'A user authentication scheme for requiring secrecy in the computer', *Comm. ACM*, Vol 17 No 8 (1974) pp 437–442
3 Harn, L, Huang, D and Laih, C S 'Password authentication based on pubic-key distribution cryptosystem', *Proc. 5th Int. Conf. Data Eng.*, Los Angeles, CA (February 1989) pp 332–338
4 Hwang, T Y 'Password authentication using public-key encryption', *Proc. Int. Carnahan Conf. Security Technology*, Zurich, Switzerland (October 1983) pp 35–38
5 Laih, C S, Harn, L and Huang, D 'Password authentication using quadratic residue', *Proc. Int. Comput. Symp.*, Taipei, Taiwan (December 1988) pp 1484–1489
6 Lennon, R E, Matyas, S M and Meyer, C H 'Cryptographic authentication of time-invariant quantities', *IEEE Trans. Comm.*, Vol 29 No 6 (June 1981) pp 773–777
7 Lamport, L 'Password authentication with insecure communication', *Comm. ACM*, Vol 24 No 11 (November 1981) pp 770–772
8 Harn, L 'A public-key based dynamic password scheme', *Symposium on Applied Computing*, Kansas City, MO (April 1991) pp 430–435
9 Chang, C C and Wu, T C 'Remote password authentication with smart cards', *IEE Proc. E*, Vol 138 No 3 (May 1991) pp 165–168
10 Chang, C C and Laih, C S 'Correspondence for remote password authentication with smart cards', *IEE Proc. E*, Vol 139 No 4 (July 1992) pp 372–372
11 Chang, C C and Hwang, S J 'Using smart cards to authenticate remote passwords', *Comput. Math. Applic.*, Vol 26 No 7 (1993) pp 19–27
12 Rosen, K H *Elementary Number Theory and Its Applications*, Addison-Wesley, Reading, MA (1988)
13 Rabin, M O *Digitalized Signatures and Public Key Functions as Intractable as Factorization*, MIT/LCS/TR-212 (January 1979)
14 Harn, L and Kiesler, T 'An efficient probabilistic encryption scheme', *Infor. Process. Lett.*, Vol 34 (April 1990) pp 44–50