An Interactive Data Exchange Protocol Based on Discrete Exponentiation

G. Agnew, R. Mullin, S. Vanstone
University of Waterloo
Waterloo, Ontario, Canada

Introduction

In the following paper, we propose a protocol for interactive data exchange. An interactive data exchange session can be divided into three phases as shown in Fig. 1:

i)a Session Key Exchange/User Authentication phase
ii)a Data Exchange Phase, and
iii)a Resynchronization phase (for error recovery).

The cryptographic system proposed for this system is based on discrete exponentiation, that is, all operations (though not shown) involve reduction modulo p for a large prime p. The security of the system is based on the difficulty of determining logarithms in a finite field GF(p) [1]. We also assume the existence of a trusted Public Key Notary (PKN). The PKN provides a certification service for each of the users' "public" keys and is not required to be on line.

Key Exchange Phase

In this phase, a session key is passed between two users. This exchange provides mutual authentication of the users involved the session and is resistant to spoofing by impersonation. The sequence begins with each user in possession of its secret exponent value (a for user A), the common modulus p, the common primitive element α and the "well-known" public key of the PKN α^{PKN}.

The PKN produces entries of the form $\left(\alpha^{-i}, S_i\right)$, for each of the network users where α^{-i} is user i's "public" key and S_i is a signed version of that key. The certificate, S_i is the pair (w,x) formed such that x is solved for the congruency

$$\alpha^{-i} = pkn * \alpha^{w} + wx$$

for a random value w (pkn is the private information of the PKN). This

C.G. Guenther (Ed.): Advances in Cryptology - EUROCRYPT '88, LNCS 330, pp. 159-166, 1988.

procedure and the key exchange protocol are described by ElGamal [2]. This is shown in Fig. 2. The procedure begins when user A initiates a call to user B (initiator/respondent respectively). The protocol proceeds as follows:

i)User A generates a random initial key K, and a random value r.
ii)User A obtains the pair $\left(\alpha^{-b}, S_B\right)$ in a public manner (e.g., from a public key directory, from B or by other means).
iii)User A verifies user B's public key by computing

$$\left(\alpha\right)^{\alpha^{-b}} = \left(\alpha^{pkn}\right)^{w} * w^{x}$$

iv)If the verification passes, user A applies the ElGamal protocol to form the message

$$\left[\alpha^{r}, \left(\alpha^{-b}\right)^{r} * K\right]$$

This is forwarded to user B along with a request for setting up a session.
v)Upon receipt, B recovers the initial session key from the message by using its secret exponent b

$$K = \left(\alpha^{r}\right)^{b} * \left(\alpha^{-b}\right)^{r} * K$$

At this point, data communications could proceed, but no authentication of User A has been performed.

vi)For mutual authentication, user B obtains $\left(\alpha^{-a}, S_A\right)$ by public means and verifies the key (as before)
vii)The actual session key is now formed as

$$K_0 = \left(\left(\alpha^{-a}\right)^{b}\right)^{K}$$

It can be seen that user A can also form this key from its secret and authenticated data. This completes the Key Exchange phase of the protocol. In the next section, we examine a "conventional" cryptographic system based on discrete exponentiation.

Data Exchange Phase

The Key Exchange phase established a common, mutually authenticated key K_0 between users A and B. From K_0, two sub-session keys K_0^I and K_0^R are derived one for each direction of data exchange (session initiator, session respondent respectively).

Before any data is exchanged, each user verifies the correct exchange of the initial keys. To do this, user A calculates the pattern

$$\alpha^{K_0^I}$$

and forwards this to B. Similarly, user B calculates the pattern,

$$\alpha^{K_0^R}$$

and forwards this to A. Each end verifies that the correct image has been received from the other user (see Fig. 3).

Once verification has been performed, the actual data exchange may begin. Ciphertext blocks are formed as

$$C_i^j = \alpha^{K_i^j} * M_i^j$$

where $j = I$ or K depending on the direction of data flow, and i indicates the message block number. The key, K_i^j used for each block is unique and is derived from the appropriate sub-session key as

$$K_i^j = f\left(K_{i-1}^j\right)$$

(this can be done in many ways). Using this technique, plus some error detection bits added to the plaintext, will allow for the detection of inserted, deleted or modified blocks.

Rendezvous Phase

The data exchange protocol will now proceed until the end of the session or until an error occurs. If an error cannot be corrected by simple retransmission, or if synchronization is lost, then a "Rendezvous" must be executed (see Fig.1). In this phase, the receiving user (B in Fig. 4) must notify the sending user that synchronization has been lost. The sender then determines the last correctly received message block (we assume that a communication protocol is present on the link to provide acknowledgments for correctly received blocks). The sender then increments the state of the key by a value n such that

$$K_i^j = K_{l+n}^j$$

where l is the last correctly received block. The sender then calculates the image

$$\alpha^{K_1^j}$$

and sends this to the receiving user. The receiving user increments its key state by an amount $n-q$ and calculates successive values of $\alpha^{K_z^j}$ until the pattern is matched (note: since synchronization has been lost, the state of either end is unknown, thus the "hunt" process must cover a sufficiently large number of exponents as to make resynchronization highly probable). Once resynchronization has been established, the data exchange phase may proceed once again.

As shown in Fig. 1 and 4, a provision has been made to try the rendezvous procedure only two times, if resynchronization is not established in this time, then the session is considered unusable and a key exchange phase is started once again. (It is also possible that the key exchange phase may fail a number of times, though not indicated, and provisions must be included to limit the number of tries for key exchange. If this occurs, then the channel must be deemed unusable.)

Conclusions

In this paper, we have described a protocol for interactive data exchange which provides strong mutual authentication of the users and data integrity. The protocols used are based on a cryptographic system using discrete exponentiation for public key exchange and conventional data exchange. The protocol is robust to data/protocol errors and active attacks. While it has been shown as an interactive protocol, a one-way data exchange protocol (for email or file transfer) can easily be derived from this protocol.

References

1. W. Diffie, M. Hellman, "New directions in cryptography", IEEE Trans. on Info. Theory, Vol. IT-22, pp.472-492, 1976.

2. T. ElGamal, "A public key cryptosystem and a signature scheme based on discrete logarithms", IEEE Trans. on Info. Theory, Vol. IT-31, pp.469-472, 1985.

Figure 1 - INTERACTIVE PROTOCOL

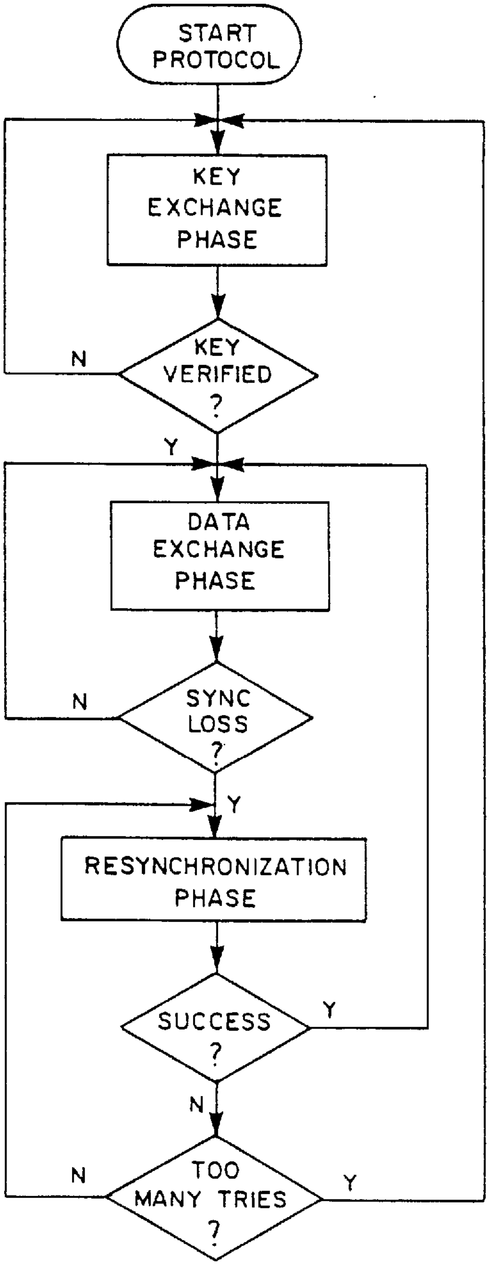


Figure 2 - KEY EXCHANGE PHASE

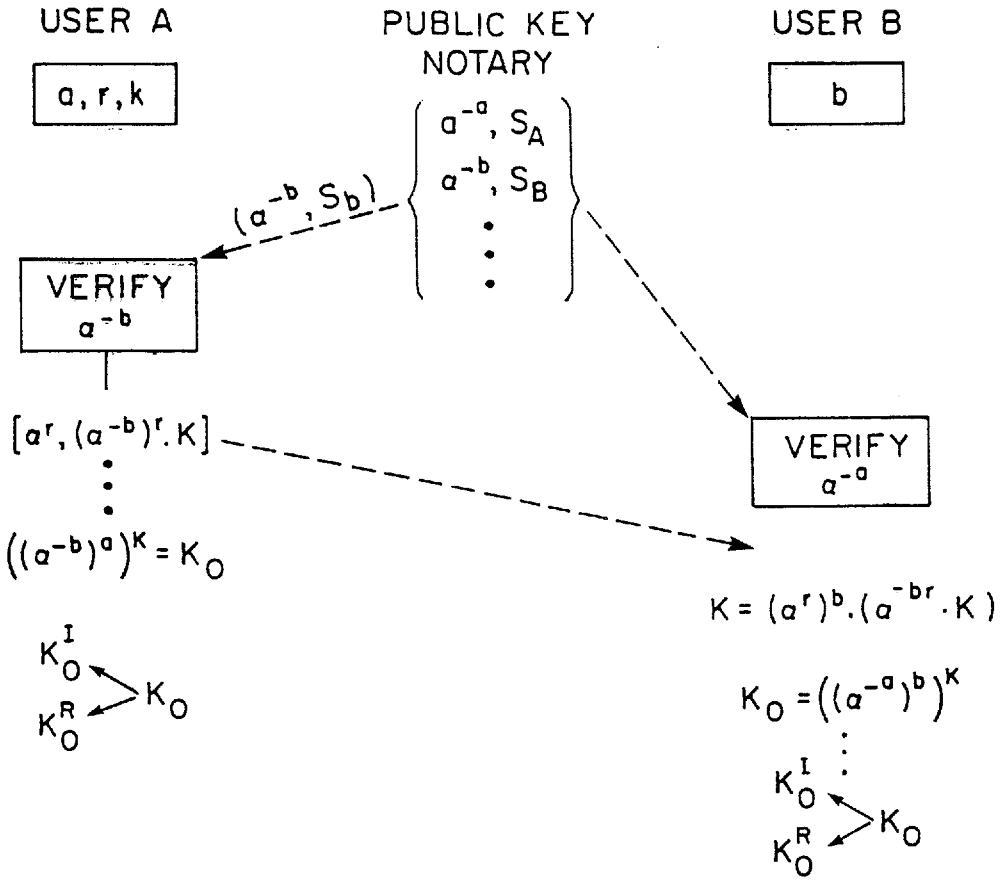


Figure 3 - DATA EXCHANGE PHASE

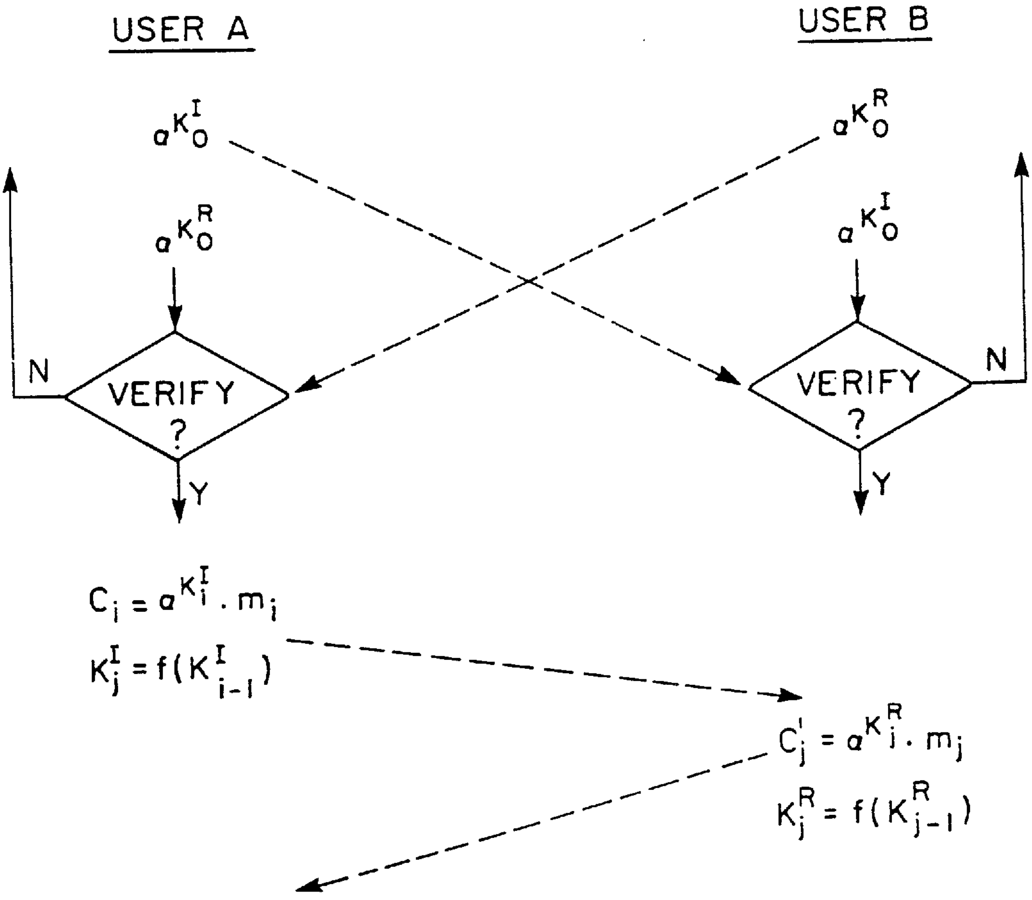
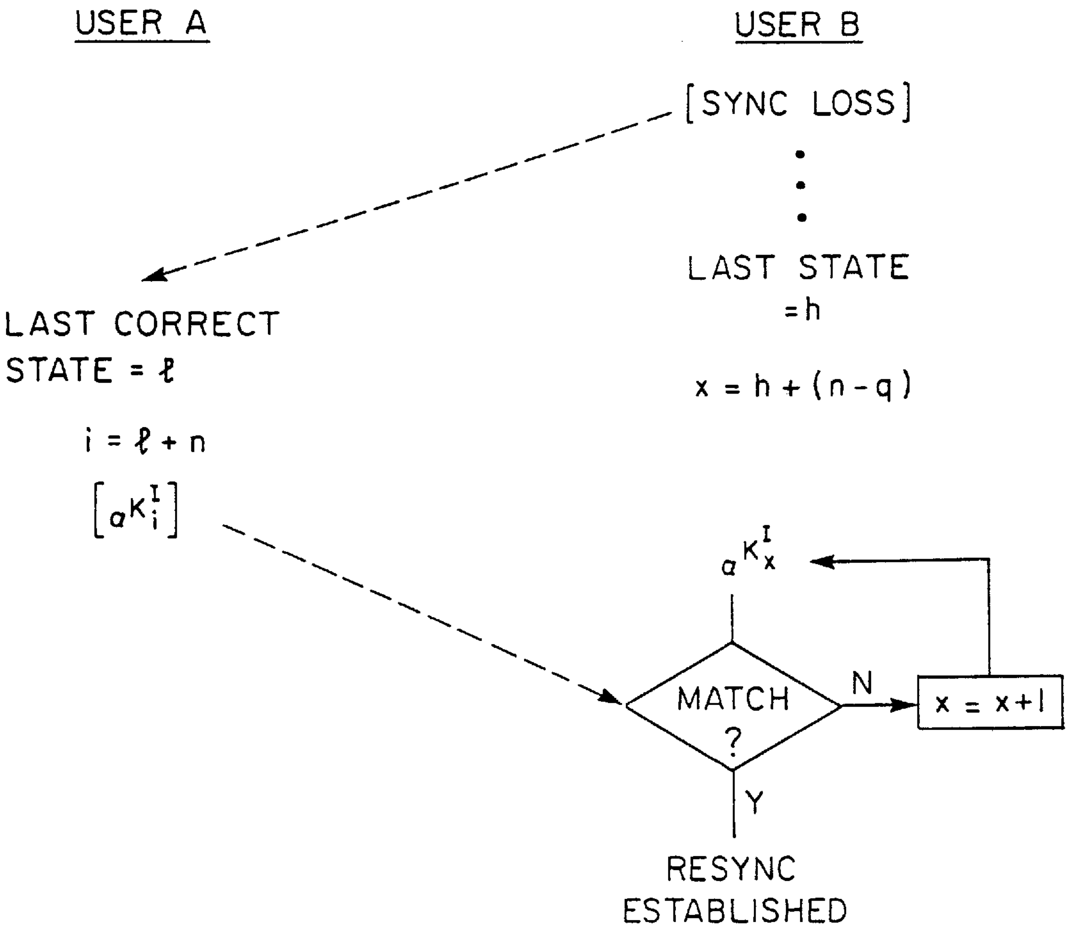


Figure 4 - RESYNCHRONIZATION PHASE


USER A
USER B
[SYNC LOSS]
LAST STATE
= h
$x = h + (n - q)$
LAST CORRECT
STATE = ℓ
$i = \ell + n$
$[{}_aK_i^I]$
${}_aK_x^I$
MATCH
?
N
$x = x+1$
Y
RESYNC
ESTABLISHED