Questo fascicolo contiene
la lezione di domani
martedì 16 aprile 2019

(Scaricatelo e portatelo in
aula F. 8)

44

MATHEMATICAL CRYPTOGRAPHY Ferrara 2003

The Massey-Omura Protocol

Mohan Nair, University of Glasgow *m.nair@maths.gla.ac.uk*

The Massey-Omura Protocol and its application to playing cards

Consider the following scenario:

- B wants to send A a message in a strong box. He places the message in it and then puts a lock on the box for which only he has a key.
- The box is then sent to A who naturally cannot open it but what she does is to put another lock on the box for which only *she* has a key. The box (now with 2 locks) is sent back to B.
- B then removes his original lock (with his key) and returns the box (now carrying only A's lock) back to A.
- Finally, A removes her lock and opens the box to receive the message.

Note that at no point is the box unlocked or there is an exchange of keys or there is any exchange of information on how to open the box.

The Massey-Omura Protocol imitates this procedure. Initially this was described in the context of elliptic curves and based on an idea of Shamir.

Two parties A and B want to communicate securely. They decide on a common large prime number p (which does not need to be secret). Then each one of them, privately and independently, chooses encryption and decryption keys e_A, d_A and e_B, d_B where

- $\mathrm{GCD}(e_A, p-1) = 1$ and $\mathrm{GCD}(e_B, p-1) = 1$ and
- $e_A d_A \equiv 1 \pmod{p-1}$ and $e_B d_B \equiv 1 \pmod{p-1}$.

The protocol for communicating a message M, $1 < M < p$, from B to A is:

- B sends $M_1 \equiv M^{e_B} \pmod{p}$ to A.
- A then calculates $M_2 \equiv M_1^{e_A} \equiv M^{e_A e_B} \pmod{p}$ and sends this to B.
- B now calculates $M_3 \equiv M_2^{d_B} \equiv (M^{e_B d_B})^{e_A} \equiv M^{e_A} \pmod{p}$ and sends this to A.
- Finally, A calculates $M_4 \equiv M_3^{d_A} \equiv M^{e_A d_A} \equiv M \pmod{p}$ to recover M.

The Massey-Omura Protocol can, of course, be used as a private key cryptosystem but it is more useful as a comparitively slow but secure method of communicating a common key between 2 individuals or entities.

Playing cards over the telephone

Now suppose two remote parties wish, for example, to play E-poker. This consists of dealing 5 random cards to each player from a full deck of 52 cards. Let A be the dealer and B the shuffler. First A encripts *all* the cards in the deck (with E_A) and sends these to B. B now shuffles the deck and also encrypts the cards (with E_B) and returns the twice-encrypted deck to A. A now removes 10 cards from the deck, decrypts 5 of them (with D_A) and sends these 10 cards back to B. B then decrypts all ten (with D_B). Five will now be visible to B, those which were decrypted by A, and this is B's hand. The remaining five are returned to A who decrypts them and this will be A's hand.

For those interested, this procedure, in detail, is as follows:

The 52 cards in a full deck are each associated with 52 unique natural numbers > 1, say $\{n_1, \cdots, n_{52}\}$. This is agreed by both players and visible to both who also choose a common large prime p. All encryption and decryption are done modulo p.

- A encrypts all 52 cards with her encrypting procedure to get $L_1 = \{E_A(n_1), \cdots, E_A(n_{52})\}$ and sends L_1 to B.

- B shuffles the cards to get $L_2 = \{E_A(m_1), \cdots, E_A(m_{52})\}$ where $(m_1, \cdots, m_{52})$ is a permutation of $(n_1, \cdots, n_{52})$. He also encrypts all the cards in L_2 with his encrypting procedure to get $L_3 = \{E_BE_A(m_1), \cdots, E_BE_A(m_{52})\}$. This is sent to A.

- A now removes 10 cards from the list and since they have been shuffled by B and invisible to both players, she might as well choose the first 10 (although this is not necessary). She obtains $L_4 = \{E_BE_A(m_1), \cdots, E_BE_A(m_{10})\}$.

However, she may not trust B's shuffle and so might prefer some other 10 arbitrary cards $L_5 = \{E_BE_A(l_1), \cdots, E_BE_A(l_{10})\}$ where $(l_1, \cdots, l_{10})$ is a 10-element sublist of $(m_1, \cdots, m_{52})$. She decrypts 5 of them and sends them all back to B. Therefore she sends $L_6 = \{D_AE_BE_A(l_1), \cdots, D_AE_BE_A(l_5), E_BE_A(l_6), \cdots, E_BE_A(l_{10})\}$. So $L_6 = \{E_B(l_1), \cdots, E_B(l_5), E_BE_A(l_6), \cdots, E_BE_A(l_{10})\}$ is sent back to B.

- B decrypts this list to get $\{l_1, \cdots, l_5, D_BE_BE_A(l_6), \cdots, D_BE_BE_A(l_{10})\}$ which is $\{l_1, \cdots, l_5, E_A(l_6), \cdots, E_A(l_{10})\}$. He takes $\{l_1, \cdots, l_5\}$ to be his cards and sends $\{E_A(l_6), \cdots, E_A(l_{10})\}$ to A who decrypts them to get $\{l_6, \cdots, l_{10}\}$ as her cards.

Note that neither A nor B knows what cards the other person has so that this procedure is, in general, fair.

4-6

MATHEMATICAL CRYPTOGRAPHY	Ferrara 2003

Tossing an e-coin

Mohan Nair, University of Glasgow *m.nair@maths.gla.ac.uk*

Suppose that a protocol involving 2 people A and B requires them to choose a sequence of bits at random. Suppose, further, that there is an advantage for A *not* to do so. For B to be sure that A does not cheat, he tosses a fair coin with A calling "heads" or "tails". If A calls correctly (and this must be clear to both people), the bit is chosen to be 1 otherwise it is 0. B repeats this process until the complete sequence of random bits is obtained. This achieves the following desired goals:

- It guarantees to B that, at each toss, A picks her bit at random.
- It guarantees to A that, at each toss, B did not know which bit he tossed to her - that he could not interfere in the procedure.

The remote coin-tossing protocol

Two people A and B want to decide something by tossing a fair coin. There are 3 stages:

- A prepares the fair "coin".
- B "tosses" the coin.
- A then "calls" - heads or tails.

Procedure

- A chooses 2 different large prime numbers p and q (both congruent to 3 (mod 4)). The coin is the number $n = pq$. A sends n to B.
- B "tosses" the coin by choosing an $a, 1 < a < n$, at random and computing $b \equiv a^2 \pmod{n}$. B sends b to A. Note that there is an extremely small chance (approximately $2/n$) that $\mathrm{GCD}(a, n) \neq 1$. If this happens, B has accidentally factorized n and wins. We can safely ignore this case!
- A's "call" consists of solving the equation $x^2 \equiv b \pmod{n}$ (there are 4 positive solutions modulo n) which she can easily do since she knows both p and q. She chooses one on these 4 solutions at random, say t, and sends t to B.

If B can now announce the values of p and q, he wins the toss. If not, A wins.

Explanation

Since B knows a (and so also $n-a$), he knows precisely 2 of the 4 solutions of the congruence $x^2 \equiv b \pmod{n}$. By sending B the value t, A gives B a 50% chance of obtaining the other 2 solutions as well. If $t \not\equiv \pm a \pmod{n}$ then $\mathrm{GCD}(t-a,n)$ will be either p or q and hence, using $n = pq$, both p and q will be known to B. This is not possible if $t \equiv \pm a \pmod{n}$.

So all B needs to do at the final step is to check if $t = a$ or $t = n - a$. If one of these is true, he has lost. If neither is true, he calculates $\mathrm{GCD}(t-a,n)$. This is either p or q and hence, using $n = pq$, he obtains both p and q and wins.

Note: A can solve $x^2 \equiv b \pmod{n}$ as follows:

$x^2 \equiv b \pmod{p}$ has solutions $x \equiv \pm r \pmod{p}$ where $r \equiv b^{\frac{p+1}{4}} \pmod{p}$.

Since $p \equiv 3 \pmod{4}$, this follows immediately from

$$(b^{\frac{p+1}{4}})^2 \equiv b^{\frac{p+1}{2}} \equiv b.b^{\frac{p-1}{2}} \equiv b \pmod{p}.$$

Note that $b^{\frac{p-1}{2}} \equiv a^{p-1} \equiv 1 \pmod{p}$.

Similarly, $x^2 \equiv b \pmod{q}$ has solutions $x \equiv \pm s \pmod{p}$ where $s \equiv b^{\frac{q+1}{4}} \pmod{q}$.

Determining c and d by the Euclidean Algorithm to satisfy $cp + dq = 1$, it is then easy to verify that the 4 solutions of $x^2 \equiv b \pmod{n}$ are precisely $\alpha, n-\alpha$ and $\beta, n-\beta$, where $\alpha \equiv rdq + scp \pmod{n}$ and $\beta \equiv rdq - scp \pmod{n}$, and that these are distinct.

Esempio numerico (banale) di

Protocollo di Massey – Omura

A e B si accordano scegliendo il primo $p = 47$ (che può essere reso pubblico).

A sceglie $e_A = 17$ e risolve la congruenza

$$e_A d_A \equiv 1 \pmod{p-1} \quad \text{cioè}$$

$$1)\quad 17 d_A \equiv 1 \pmod{46}$$

Applica quindi l'algoritmo euclideo

$$46 = 17 \cdot 2 + 12$$
$$17 = 12 \cdot 1 + 5$$
$$12 = 5 \cdot 2 + 2$$
$$5 = 2 \cdot 2 + 1$$

da cui segue

$$1 = 5 - 2 \cdot 2 = 5 - 2(12 - 5 \cdot 2) = 5 \cdot 5 - 2 \cdot 12 =$$
$$= 5(17 - 12) - 2 \cdot 12 = 5 \cdot 17 - 7 \cdot 12 =$$
$$= 5 \cdot 17 - 7(46 - 17 \cdot 2) = 19 \cdot 17 - 7 \cdot 46$$

Perciò la minima soluzione positiva d_A della congruenza 1) è $d_A = 19$. Ne segue

$$2)\quad \begin{cases} e_A = 17 & \text{(encryption-key di A)} \\ d_A = 19 & \text{(decryption-key di A)} \end{cases}$$

B a sua volta sceglie $e_B = 11$ e risolve la congruenza ②

$$e_B d_B \equiv 1 \pmod{p-1}, \text{ cioè}$$

3) $$11 d_B \equiv 1 \pmod{46}$$

Si ha

$$46 = 11 \cdot 4 + 2$$
$$11 = 2 \cdot 5 + 1$$

da cui segue

$$1 = 11 - 2 \cdot 5 = 11 - (46 - 11 \cdot 4) \cdot 5 = 11 \cdot 21 - 5 \cdot 46$$

Perciò la minima soluzione positiva della congruenza 3) è $d_B = 21$. Ne segue

4) $$\begin{cases} e_B = 11 & (\text{encryption-key di } B) \\ d_B = 21 & (\text{decryption-key di } B) \end{cases}$$

A questo punto B vuol mandare ad A il messaggio

5) $$M = 7$$

(cioè mette il suo "lucchetto")

Calcola quindi $M_1 \equiv M^{e_B} \pmod{p}$, per noi (vedi 4))

6) $$M_1 \equiv 7^{11} \pmod{47}$$

Si ha $7^1 \equiv 7 \pmod{47}$
$7^2 \equiv 49 \equiv 2$
$7^4 \equiv 4$
$7^8 \equiv 16$

e quindi $7^8 \cdot 7^2 = 7^{10} \equiv 16 \cdot 2 = 32 \equiv -15 \pmod{47}$

da cui $7^{11} \equiv (-15) \cdot 7 = -105 \equiv 36 \pmod{47}$

Quindi B manda ad A

$$7) \quad M_1 = 36$$

A non può decifrare ed aggiunge il suo "lucchetto", cioè calcola (vedi 2) e 7))

$$8) \quad M_2 \equiv M_1^{e_A} \equiv (36)^{17} \equiv (-11)^{17} \pmod{47}$$

Si ha

$$11 \equiv 11 \pmod{47}$$
$$11^2 = 121 \equiv -20$$
$$11^4 \equiv 400 \equiv -23$$
$$11^8 \equiv 23^2 = 529 \equiv 12$$
$$11^{16} \equiv 144 \equiv 3$$

e infine $(-11)^{17} \equiv -3.11 = -33 \equiv 14 \pmod{47}$

Perciò A manda a B

$$9) \quad M_2 = 14$$

A questo punto B toglie il suo "lucchetto" calcolando (vedi 4) e 9))

$$10) \quad M_3 \equiv M_2^{d_B} \equiv 14^{21} \pmod{47}$$

Si ha

$$14 \equiv 14 \pmod{47}$$
$$14^2 \equiv 196 \equiv 8$$
$$14^4 \equiv 64 \equiv 17$$
$$14^8 \equiv 289 \equiv 7$$
$$14^{16} \equiv 49 \equiv 2$$

da cui $14^{20} \equiv 2.17 = 34 \pmod{47}$

e infine $14^{21} \equiv 34.14 = 476 \equiv 6 \pmod{47}$

Quindi B manda ad A

$$11) \quad M_3 = 6$$

Per finire A toglie il suo "lucchetto" calcolando

$$12) \quad M_4 \equiv M_3^{d_A} \equiv 6^{19} \pmod{47}$$

(vedi 2) e 11)).
Si ha (vedi calcoli pagina precedente)

$6 \equiv 6 \pmod{47}$
$6^2 \equiv 36 \equiv -11$
$6^4 \equiv 121 \equiv -20$
$6^8 \equiv 400 \equiv -23$
$6^{16} \equiv 23^2 \equiv 529 \equiv 12$

da cui $6^{18} \equiv (-11)(12) \equiv -132 \equiv$
$\equiv 9 \pmod{47}$

e infine
$6^{19} \equiv 6.9 = 54 \equiv 7 \pmod{47}$,

Si ha perciò

$$13) \quad M_4 = 7$$

Come si può vedere (vedi 5) e 13)) è stato recuperato il messaggio di partenza, $M = 7$.

—

<u>Osservazione</u>

Se Oscar riesce ad intercettare tutti i messaggi che si scambiano Bob e Alice conosce $M_1 = 36$, $M_2 = 14$, $M_3 = 6$: da questi non riesce a risalire al messaggio vero (che non viene mai trasmesso senza almeno un lucchetto) $M_4 = M = 7$.