6.857 Computer and Network Security **Fall Term, 1997**

Lecture 3 : September 11, 1997

Lecturer: Ron Rivest *Scribe: Abby Knickerbocker*

1 Unconditionally Secure Authentication

The key issue addressed in this lecture is that of *authentication*; how can Bob be sure that the message he has received was actually sent by Alice? In addition, how can he be sure that the message was not corrupted en route by Eve?

1.1 Message Authentication Code (MAC)

The idea here is to send the message M along with a MAC, $f(M,k)$, that is a function of the message M and the secret key k. How does the MAC allow Bob to assure himself that the message M is really from Alice? Alice and Bob share the secret key k. Bob recomputes the MAC and confirms that the message M was from Alice.

Note: Authentication is a separate goal from privacy; here, we are not concerned with confidentiality. M is sent cleartext over the channel, and we do not care who hears it.

It is also important that a new key k is used every time. For instance, this prevents replay attacks. Eve cannot resend the message the next day; Bob will not accept it.

GOAL: Eve should not be able to generate a valid pair $M^{'}, f(M^{'}, k)$ even after hearing one valid pair $M, f(M,k)$.

IDEA: The one-time key k is a pair (a,b) of coefficients: $k = (a,b)$ and then we define $f(M,k) = y = aM + b$. Having seen one (M,y) pair is of no help in figuring out the appropriate $y^{'}$ for a different message $M^{'}$.

More formally: Suppose $|M| = n$ (i.e. M has n bits).

Let p be a prime, with $|p| > n$.

Then $Z_p = \{0, 1, ..., p-1\}$ forms a field in modulo p with respect to $+, -, *, /$.

- $+$ is associative and commutative with identity 0

- $*$ is associative and commutative with identity 1
- all elements have additive inverses: $-5 = 12 (mod 17)$
- all nonzero elements have multiplicative inverses: $5^{-1} = 7 (mod 17)$
- division: $2/5 = 2 * 7 = 14 (mod 17)$ since $5^{-1} = 7$

Now choose a and b uniformly at random from Z_p, and define $f(M, k) = aM + b(mod p)$.

Note: $|f(M, k)| = |p|$, so the MAC is as long as the message.

Suppose Eve hears (M, y), where $y = aM + b(mod p)$. What has she learned? She knows M, y, and p, but not a and b. There are p pairs of coefficients (a, b) that are consistent with the equation $y = aM + b(mod p)$. Suppose Eve picks some $a_i \in Z_p$. Then $b_i = y - a_i M(mod p)$ satisfies the above equation. All such choices (a_i, b_i) are equally likely, as far as Eve knows.

Suppose Eve wants Bob to accept a different message $M' \neq M$. She must compute a new MAC that Bob will accept.

$f(M', k) = a_i M + (y - a_i M)(mod p) = a_i(M' - M) + y(mod p)$

where a_i is Eve's guess. BUT, these give different values for different choices of a_i, since:

$a_i(M' - M) + y = a_j(M' - M) + y$

$a_i = a_j$

Thus, $f(M', k)$ is equally likely to be any value in Z_p. Therefore, Eve's chance of getting $f(M', k)$ right is $1/p$.

One-time Key k

What happens if Alice and Bob use the same secret key k to compute the MAC's for two separate messages M and M'? Then they are computing the two equations $y = f(M, k)$ and $y' = f(M', k)$. This gives the two following equations:

$y = aM + b(mod p)$

$y' = aM' + b(mod p)$

But that gives Eve two equations in two unknowns, and she can solve for a and b. $a = (y - y^{'})/(M - M^{'})$ and $b = y - aM$. Then Eve knows both a and b! Therefore, **do not** reuse a key.

Length of the MAC

Both the key and the MAC have length proportional to $|M|$. Having $|k| = |M|$ is unavoidable. However, the length of the MAC is long, and it would be nice if we could improve this.

Ideas to shorten MAC:

- only send low-order 64 bits of $y = aM + b(modp)$
- choose small p (64, 65 bits)

  divide M into chunks $M = M_1, M_2, ..., M_t$ where $0 <= M_i < p$

  $k = (a_1, a_2, ..., a_t, b)$ where $0 <= a_i < p$, and $0 <= b < p$

  $f(M, k) = \sum a_i M_i + b(modp)$

1.2 Privacy and Authentication

Previously, we were only concerned with authentication. Alice was sending her message M cleartext over the channel, and Eve could read it. Now we want to consider the case where Alice wants her message to be private so that only Bob can read it, *and* she wants to authenticate the message. Now we combine the One-Time Pad (OTP) and the One-Time MAC (OTM). We have two choices:

- encrypt M with OTP, *then* append MAC of ciphertext: $M \oplus k, f(M \oplus k, k^{'})$
- append MAC to M, use OTP to encrypt (M, y) pair: $(M, f(M, k^{'})) \oplus k$

There are two advantages of the first choice, where we encrypted the message M and then appended the MAC of the ciphertext. First of all, Bob can check the MAC before decrypting. If it is garbage, then he does not have to waste time decrypting $M \oplus k$. Second, encrypting the message first uses fewer random bits off the pad:

- $M \oplus k, f(M \oplus k, k')$

  $|k| = n$

  $|k'| = n$

  $n + 2n = 3n$

- $(M, f(M, k')) \oplus k$

  $|k'| = 2n$

  $|k| = 2n$

  $2n + 2n = 4n$

1.3 Length of One-Time Key k

Is it possible to shorten the length of the one-time key k, $|k|$, needed for privacy or authentication? To do so, we must start making assumptions about computational difficulty.

Alice encrypts a message M that is n bits long with a key k that is t bits long, resulting in ciphertext C, which is also n bits long. Imagine that Eve hears C and has an infinite amount of computing power. Eve tries all 2^t keys and examines the resulting plaintext M. She then gets 2^t candidate messages. Assume that the number of reasonable messages of length M is α^n $(1 <= \alpha < 2)$ (e.g. English $\alpha = 1.1$). The expected number of keys giving a reasonable result is thus:

$2^t(\alpha/2)^n$

Thus, when $t + n(lg(\alpha) - 1) < 0$ then the number of keys giving reasonable results $<= 1$. When:

$n > t/(1 - lg(\alpha))$

then there is only one key that works ($n > 7t$ for English). This argues that unconditional security requires growing the length of the keys with the length of the message. Thus, unconditional security requires *long* one-time keys.