Lecture 5 - CSE 5852

Tham Hoang

September 19, 2016

1 Recap

Last class we introduce the concept of a message authentication code. This was designed to detect when an adversary actively changed the message being sent. We introduced the MAC experiment $\mathtt{Mac} - \mathtt{forge}^{\mathcal{A},\mathsf{Mac}}$:

Crytosystem		**Attacker**
Choose α	$- - - - - - - \rightarrow$	
	$\leftarrow - - - - - - m$	
$t = \mathsf{Mac}(\alpha, m)$	$- - - \rightarrow$	
	$\leftarrow - - - - - - (m', t')$	

Attacker wins if $m \neq m'$ and $\mathsf{Vfy}(\alpha, m', t') = 1/True$
We say that a Mac is secure if for all attackers, $Pr[Attacker\ wins] \leq \epsilon$.

2 Hash Function

Strategy For the first time rather than directly constructing an object we will introduce another (simpler) cryptographic object and show it is sufficient to build a good Mac. After, we will instantiate this simpler object. However, this will require a bit of number theory and will take some time. This will be our process throughout much of the class, showing how to use simpler objects to build more complicated and powerful cryptography.

What should be true about t knowing m, m', t?
- Hard to guess/ lots of values
- Independence (based on perfect secrecy)

Goal: Build a function where each output is random and knowing one output doesn't tell about another.

Definition 1. *A hash func, $h : K \times C \rightarrow T$ is strongly universal for all pairs m, m' where $m \neq m'$ and for all $t, t' \in T$*

$$Pr_{\alpha}[h(\alpha, m) = t \cap h(\alpha, m') = t'] = \frac{1}{|T|^2}.$$

Output is random for each pair of inputs. This type of function is also known as pairwise independent. How is this different than a secure MAC?

1) The choice of m' does not depend on t.
2) No interaction.
3) No attacker.
4) One func h instead of 2 (MAC and $Verf$)

MAC construction

Choose α.
$t = \mathsf{Mac}(\alpha, c) = h(\alpha, c)$,
$\mathsf{Vfy}(\alpha, c, t) =$
1) $t' = h(\alpha, c)$
2) Output 1 if $t' = t$
Remember that MAC scheme is to prevent the attacker to change ciphertext.

Theorem

This construction is an $\epsilon - unforgeable\ MAC$ for $\epsilon = \frac{1}{|T|}$

Proof

Consider some A. A outputs m. Consider some fixed m_1, m_2 either

$$\Pr[A\ wins\ when\ outputs\ m_1] \geq \Pr[A\ wins\ when\ outputs\ m_2]$$

or the opposite is true. Without loss of generality, we can consider an adversary that only outputs the m where their success probability is maximal. Thus, we can assume that A outputs a fixed message.
There exists A' that is as successful and always outputs the same m. After that, the Attacker sees $t = h(\alpha, m)$. They then outputs some m', t' after seeing t. By the same argument we can assume that m', t' are deterministic function of t.

Thus, we can consider the probability that A wins the game as follows:
$= \sum_t Pr[Adv\ wins \cap h(\alpha, m) = t]$
$= \sum_{t, m', t' = A(t)} [Pr[h(\alpha, m') = t' \cap h(\alpha, m) = t]]$ for fixed m.
$= \sum_t \frac{1}{|T|^2}$
$= \frac{1}{|T|}$

We will quickly introduce a strongly universal hash function and then take a detour into number theory to cover the necessary details to prove this construction is secure.

Theorem 1. *$h(a, b, c) = ac + b\ mod\ p$ is strongly universal for p prime*

3 Number Theory

To verify the above theorem we'll need to introduce a bit of number theory and get comfortable working with groups.

We will be working with different group structures in this class. Some of these structures may look familiar and be represented by integers. It is very important not to assume structure that we don't talk about. For example, don't assume you can divide two elements. Looking ahead, the basis of much of cryptography is that certain operations are "difficult." Thus, we want to be very careful about what operations are 1) possible and 2) computationally efficient. We will explicitly introduce what is possible. If you need to use some other operation/property, you need to show or state that it holds true. To deal with this transition, take everything slow and question whether it is possible to perform each operation.

Groups

A set G and an operation $\times$

1) $\forall a, b \in G \rightarrow a \times b \in G$ (Closure)

2) $\forall a, b, c \in G \rightarrow a \times (b \times c) = (a \times b) \times c$ (Associativity)

3) There exists $e \in G$ such that $\forall b \in G \rightarrow e \times a = a = a \times e$ (Identity)

4) $\forall a \in G$, there exists $b \in G$ such that $a \times b = b \times a = e$ (Inverse)

b is often called a^{-1}

Examples:

Real numbers with addition is a *group*

Real numbers with multiplication is not a *group* because Inverse does not satisfied for zero.

Real number without zero is a *group.*

Integer numbers with addition is a *group*

Integer numbers with multiplication is not a *group.*

Note: When the group operation naturally corresponds to our notion of "addition" we will instead write the group operation as $+$.

3.1 Modular Arithmetic

E.x, 243 = 79 * 3 + 6 $\rightarrow$ remainder 6. For any a, b can write $b = aq + r$ where $0 \leq r \leq a$ (this also hold with negative values). We say $a|b$ or divides b if $r = 0$ and $a \nmid b$ if $r \neq 0$.

Theorem 2. *If $a|b$ and $a|c$, then for any x, y, $a|(xb + yc)$.*

Proof. If $a|b \rightarrow b = aq_1$

$a|c \rightarrow c = aq_2$

$\rightarrow bx + cy = aq_1x + aq_2y = a(q_1x + q_2y)$

$\rightarrow a|bx + cy$ □

If a is positive and $a|b$ then is called a divisor of b. A number p is prime if only divisors are 1 and p. Other positive integers are composite.

Theorem 3. *For any a, b where b is positive, there exists unique a, r such that $b = aq + r$ and $0 \leq r \leq a$*

Furthermore, it is possible to efficienctly find q, r.

3.2 Greatest Common Divisor

$\gcd(a, b)$ is largest c such that $c|a$ and $c|b$. Some examples:

a	b	gcd(a,b)
21	4	1
729	18	9
324	81	81

Denote by r (in $b = aq + r$), we will write $b \mod a$ as shorthand for r.

b	a	b mod a
37	10	7
3	102	3
-10	21	11 (because $-10 = -1 * 21 + 11$)
21	9	3

Theorem 4. *Let $a, b > 1$ and $b \nmid a$, then* $\gcd(a, b) = \gcd(b, a \mod b)$

if $b \nmid a$
gcd(2991, 3772)

a	b	$a \mod b$
3772	2991	861
2991	861	328
861	328	123
328	123	82
123	82	41
82	41	0

The algorithm stops at this point because $b|a$.

Proposition 1. *Let $a, b > 1$ and $b|a$ then* $\gcd(a, b) = b$.

This is Euclid's Algorithm 300 BC