CSE 5852, Modern Cryptography: Foundations	Fall 2016
Lecture 6, 2016	
Prof. Benjamin Fuller	*Scribe: Lowen Peng*

1 Overview

We will quickly look at last class' material, look into some algebraic properties of $\mathbb{Z}_n$, construct a universal hash function using these properties, and (very briefly) discuss what happens when the key space is smaller than the message space, i.e., $|\mathcal{K}| < |\mathcal{M}|$.

2 Last Class

Definition 1. *A **strongly universal hash function** is a hash function*

$$h : \mathcal{K} \times \mathcal{M} \mapsto T$$

where $\forall c' \neq c, t, t'$, we have

$$\Pr[h(\alpha, c) = t \wedge h(\alpha, c') = t'] = \frac{1}{|T|^2}$$

Definition 2. *A **group** is a pair $(G, \cdot)$ where G is a nonempty set equipped with an operation $\cdot$ satisfying the following requirements:*

- *$\exists e \in G$ such that $\forall g \in G$ we have $eg = ge = g$ (existence of identity element)*
- *$\forall g, h \in G$ we have $gh \in G$ (closure)*
- *$\forall g, h, k \in G$ we have $(gh)k = g(hk)$ (associativity)*
- *$\forall g \in G, \exists g^{-1}$ such that $g^{-1}g = gg^{-1} = e$ (existence of a unique inverse)*

3 Algebra in $\mathbb{Z}_n$

Definition 3. *We write $a \bmod n$ to mean the remainder of dividing a by n. That is, if for some integer q and r where $0 \leq r < n$ we have $a = qn + r$, we can write*

$$r = a \bmod n$$

Some examples:

- $50 \bmod 37 = 13$
- $20 \bmod 37 = 20$
- $87 \bmod 37 = 13$

Definition 4. *We say a and b are **congruent**, or a is congruent to b, when $a \bmod n = b \bmod n$. This is written $a \equiv b \bmod n$.*

Observation 5. *We can check that $\equiv$ is an equivalence relation:*

- *For all a we have $a \equiv a \bmod n$ (reflexive)*
- *For all a, b we have $a \equiv b \bmod n \iff b \equiv a \bmod n$ (symmetric)*
- *For all a, b, c, if $a \equiv b \bmod n$ and $b \equiv c \bmod n$, then $a \equiv c \bmod n$ (transitive)*

Some basic arithmetic works with this relation. It can be verified that $(a+b) \bmod n \equiv (a \bmod n) + (b \bmod n)$. Similarly we have $(ab) \bmod n \equiv (a \bmod n)(b \bmod n)$. Division does not, in general, work this way. That is, for any a, b we don't necessarily have $(a/b) \bmod n = (a \bmod n)/(b \bmod n)$.

Two examples:

- $10371092 \cdot 10401 \bmod 100 = (10371092 \bmod 100)(10401 \bmod 100) = 92$
- $768 \cdot 21 \bmod 6 = (768 \bmod 6)(21 \bmod 6) = 0$

It is important to know that, if $ac \equiv bc \bmod n$, then it is **not** necessarily the case that $a \equiv b \bmod n$. In particular, c has a multiplicative inverse in integers $\bmod n$ if and only if $\gcd(c, n) = 1$.

Proposition 6. *The set $\{0, \ldots, n-1\}$ equipped with the operation addition $\bmod n$ is a group. This group is called $\mathbb{Z}_n$.*

- *$\mathbb{Z}_n$ has the identity 0*
- *For any $a \in \mathbb{Z}_n$ we have the unique inverse $n - a$*
- *Closure follows from reduction $\bmod n$*
- *Associativity inherits from the associativity of $+$*

Note that $\mathbb{Z}_n$ equipped with multiplication $\bmod n$ is not always a group.

To represent $a + a$, we can write $2a$. Similarly, $a + a + a = 3a$ and so on. Note that 2 or 3 are not necessarily in the group, such as if a is an element in a matrix group. More generally, the expression $aaa \ldots aaa$ can be written with powers, like $aaaa = a^4$, since group operations are not always represented by $+$.

Definition 7. *We write $|G|$ to denote the order of G, that is, the size of the set.*

For example, $|\mathbb{Z}_n| = n$.

It is possible to form a multiplicative group on a subset of $\mathbb{Z}_n$. Let p be a prime, then equip the set $\mathbb{Z}_p \setminus \{0\}$ with multiplication mod$p$. Since p is prime, the gcd of any number in $\{1, \ldots, p-1\}$ and p is 1. Thus, every element has an inverse, and this set-operation pair forms a group. This is denoted as $\mathbb{Z}_p^*$.[1]

4 Universal Hash Function

Construct a hash function

$$h : \mathbb{Z}_p^2 \times \mathbb{Z}_p \to \mathbb{Z}_p$$

such that $h(a, b, c) = ac + b \bmod p$. The function essentially:

- Multiplies a and c, and reduces the product mod$p$; call this new term $d = ac \bmod p$
- Adds d to b in $\mathbb{Z}_p$, so $H(a, b, c) = b + d \bmod p$

Theorem 8. *The above function is a strongly universal hash function.*

Fix $c, t, c', t' \in \mathbb{Z}_p$ such that $c \neq c'$. We want to know for which a, b do we have $ac + b \equiv t \bmod p$ and $ac' + b \equiv t' \bmod p$.

First rewrite first expression as $b \equiv t - ac \bmod p$ and use this relation in the other expression to obtain $ac' + t - ac \equiv t' \bmod p$. Some algebra lets us rewrite this as $a(c' - c) = (t' - t) \bmod p$. Observing now that $c' - c \neq 0$ and $c' - c \neq p$, we have $\gcd(c' - c, p) = 1$. Thus $a = (t' - t)(c' - c)^{-1} \bmod p$, showing the uniqueness and existence of a. Substituting this back into any of the initial expressions will also show the uniqueness and existence of b.

Since the choices of a, b given c, c' and t, t' are unique. There is a single a, b that produces c, c', t, t' and this occurs with probability $1/(|a||b|)$ or equivalently $1/|p|^2$. That is,

$$\Pr[h(a, b, c) = t \wedge h(a, b, c') = t'] = \frac{1}{|\mathbb{Z}_p|^2}$$

Two things to note.

- Did not exclude the possibility of $a = 0$ so that nothing in the tag space would be excluded
- The key is twice as long as the message

So with the above universal hash function, we can consider a one-time pad and one-time MAC as follows (for some p that is larger than the message space):

[1] Similarly we can also define a multiplicative group on $\mathbb{Z}_n$, called $\mathbb{Z}_n^*$. This group consists of all elements a where $gcd(a, n) = 1$. You should verify for yourself that multiplying two elements a, b with $\gcd(a, n) = 1$ and $\gcd(b, n) = 1$ yields an element $\gcd(ab \mod n, n) = 1$.

1. Generate the keys k, a, b
2. Get ciphertext $c = k \oplus m$, where m is the message
3. Get tag $t = ac + b \bmod p$
4. Send c, t

The above scheme is perfectly secure and unforgeable. To verify, the receiver computes $t' = ac + b \bmod p$. If $t' \neq t$, then the receiver aborts. Otherwise, the receiver computes $m = c \oplus k$ to retrieve the message.

5 What happens when the key space is smaller than the message space?

What was outlined in the previous section is information-theoretic security, where the adversary has unbounded computational power. We showed it was necessary that the key space is larger than or equal to the message space. What happens if we violate this assumption? If the key space is smaller than the message space, the image of any two messages under Enc over the support of $\mathcal{K}$ cannot always be the same set.

Attack 1 Assume that $|\mathcal{M}|$ is uniformly distributed. For a particular ciphertext c the adversary can just exhaustively check all keys and rule out the messages that do not appear. This violates perfect secrecy.

Attack 2 If the adversary knows that the hiddentext is one of two possible messages, then one manner of attack is to use their knowledge of c to guess the message by randomly generating $\mathcal{K}$ and computing $m = \mathrm{Dec}(c, \mathcal{K})$. There is a nonzero chance of determining the message, since the limited key space introduces a bias on $\mathcal{M}|\mathcal{C}$. This violates perfect secrecy.

The the former attacker has the runtime $\mathcal{K}$ and completely eliminates messages, the latter is efficient (just guess a key). This shows if we want to work in regimes when $|\mathcal{K}| < |\mathcal{M}|$ we need to adapt our definition in two ways:

1. We need to restrict to adversaries running in limited time.
2. We need to allow the adversary some probability of learning about messages.