Lecture 6

Last lecture Review:
- CBC, CFB, CTR, IND, OPA

Hash Function: Notes:
  - CR2: Strong Collision Resistance
  - CR1: Weak Collision Resistance
  - CR0

In CR2 - key given to Adv. If we ask Adv to give $x_1, x_2$ note he give same Hash.

In CR1, we find $x_1 \neq H(x_1)$ and we now we give key to Adv and ask them to find $x_2$ given same $H, C(x)$.

In CR0: No key given.

CR2 $\Rightarrow$ OW.

If $\frac{|R_1|}{|D|} \leq \epsilon$, then it holds true. Domain is large if range is small.

For example: SHA-1: $|R1| = 160$

In general we want: $\frac{|R1|}{|D|} \leq 2^{-80}$ so $|D| \geq 2^{80} |R1|$

Theorem: If \( h \in CR2 \), then \( H \in CR2 \)

Mathematically: \( \forall h \in CR2 \Rightarrow H \in CR2 \)

**Property of Pad Function:**
Take the meg & do padding on it.

1. If length of 2 megas is same, then the pad is going to be the same.
   - If \( |M1| = |M2| \)
     - Then pad \( (M1) = \text{pad}(M2) \)

2. If \( |M1| \neq |M2| \)
   - Then at least the last block of pad \( (M1) \) will be different from the last block of pad \( (M2) \).

**Proof:** If attacker can break the \( CR2 \) property of \( H \), then there exists an attacked \( H \) which breaks the \( (h \in CR2) \) property.

\[ H \in CR2 \Rightarrow \exists H \in CR2 \]
So from $x_1, x_2$ we find 2 msgs such that:

$$h_k(m_1) = h_k(m_2) \land (M_1 \neq M_2).$$

Case 1: length of $(M_1, M_2)$ returned by $A$ is not same, i.e., $|x_1| \neq |x_2|$

so from property of pad we know that at least last block is not same.

so we know:

$$h(x_1m, c_{m-1}) = h(x_2m, c_{2m-1})$$

from padding property we know these 2 msgs are collected.

so return $M_1 = (x_1m, c_{m-1})$

$M_2 = (x_2m, c_{2m-1})$.

If lengths are the same:

so we look at the 2nd last block:

$$h(x_1m, c_{m-1}) \land h(x_2m, c_{2m-1}).$$

so subcase: if $(x_1m, c_{m-1}) \neq x_2m, c_{2m-1})$

then $M_1 = (x_1m, c_{m-1})$ $M_2 = (x_2m, c_{2m-1})$. 
If \((x_1 m, (1 m-1)) = (x_2 m, (2 m-1))\), then it's the same msg. Then we don't do anything; we go to power block.

Here advantage on the same \(Adv(E) = Adv(x)\).

So general condition
\[
\begin{align*}
(x_1 m-i, c_1 m-i-1) &\neq (x_2 m-i, c_2 m-i-1) \\
\Rightarrow M_1 &= (x_1 m-i, c_1 m-i-1) \\
M_2 &= (x_2 m-i, c_2 m-i-1).
\end{align*}
\]

Message Authentication.

We were talking only about confidentiality.

Now we want Authentication.

It means if Alice sent a msg, then Bob should know Alice actually sent it.

So Bob wants to find if indeed Alice sent msg \(m\) that the integrity of the msg was not tampered with.

\[
\begin{align*}
&\text{Enc}(x) = \text{Enc}(m) = \text{Enc}(k) \\
&\text{Enc}(m) = \text{Enc}(k).
\end{align*}
\]
Use OTP. (For \( x \))

\[ m, e, k \]

\[ e = m + k \]

\[ e_1 = m_1 + k \]

\[ e + e_1 = m + m_1 \]

\[ e_1 = m_1 + e \]

\[ e_1 = k + m_1 \]

So in general: Encryption ≠ Authentication.

Message Authentication Scheme:

Key Gen: Picks a Random \( k \in \mathbb{K} \).

Outputs \( k \in \mathbb{K} \)

MAC Generation: On input \( m \) & a message \( m \in \mathbb{K} \).

It outputs a tagged MAC (MAC = MAC\(_k(\mathbb{M})\)
(2) MAC Generation can be Randomized or even Deterministic.

Unlike encryption algo's which can't be Deterministic and is insecure against Chosen Plaintext Attack.

(3) MAC Verification: Verifies the correctness of this tag (MAC), using a key on message m, outputs either "1" or "0".

Security Notes on (MAC):

(1) Key Recovery: If someone does not know the key, then they should not be able to construct a MAC. (Weak notion)

(2) No Message Attack: we are just listening to (A & B) and do not even know any (MAC) exchange. You are asked to come up with a (m, MAC) pair. (Valid one)

Even without a key can u come up with a (msg & MAC) pair?

(3) Strongest Notion: Existential forgery under adaptively chosen message attack, (CMA).
Adversary has access to the signing oracle and can generate a MAC \( \text{MAC}_1 \) on input \( M \) to get \( \text{MAC}_1 \).

Adversary can give \( \text{MAC}_1 \) to oracle and depending on \( \text{MAC}_1 \) it can modify \( M \) to get some \((M, \text{MAC}_1)\).

The adversary now needs to verify these so it sends this to the verifying oracle and if it is a valid \((M, \text{MAC})\) pair then adversary has won.

Why we need this?

Adversary is recording this communication so if \( \text{Adv} \) can come up with check \( (100,000, \text{MAC}) \) and \((1000, \text{MAC}) \) corresponding \((\text{MAC})\) value, then it's a forgery.
Theorem: E-PRF yields a MAC.

If a MAC scheme instantiated with a PRF is a secure MAC scheme under (CMA), then E-MAC is (CMA) secure.

Proof: We use Contradiction.

If E is a PRF, then E-MAC is (CMA) secure.

So if there exists an attacker who can break E-MAC, CMA security then he can break E-PRF.

1. E-MAC \rightarrow \text{(CMA Sec.)} \rightarrow \text{1 E-PRF Sec.}

A asks signing oracle to give MAC for m_i.
B will capture it and send it to challenge.
Challenge sends g(m_i)

A does it for say g times.

Now A sees m_i MAC_i pairs and then tries to get a forged msg.

so it sends forge \((m_i, \text{MAC}_i)\) msg for verification.

Now B has to simulate it again.
We say $Adv_B$ gains

This means one of the $q_V$ queries made match to the $Mac_i = g(m_i)$ for $B$ to fail.

If size of $PRF$ is $2k$ and $q_V \leq 2k$

\[ Adv_B = Adv_A - \frac{q_V}{2k}. \]
So it means:
\[ \text{Adv}(A) = \text{Adv}(B) + \frac{q_v}{2^k} \]

We want this to be negligible.
\[ \text{Adv}(B) \] is a PRF, so its adv is 0 but, the factor is increased because of \( \frac{q_v}{2^k} \)

so we want this to be 0 too.

So, we should limit the number of queries:
\[ \frac{q_v}{2^k} \leq \frac{1}{2^{60}} \]
\[ q_v \leq \frac{2^k}{2^{60}} \]
\[ q_v \leq 2^{k-60} \]

So we can use any PRF as a MAC.
Theorem: CBC-MAC

We don't need IV, because we do not need MAC to be random.

Birthday Attack on CBC MAC:

\[ \text{Adv} \text{ has access to signing \& verification oracle.} \]

Adversary queries \( (m_{11}, m_{12}, a, \ldots, a) \)
\( (m_{21}, m_{22}, o, \ldots, a) \)
\( (m_{31}, m_{32}, a, \ldots, a) \)
\( (m_{41}, m_{42}, a, \ldots, a) \)

So these differ only on set 2 blocks.

So what \( \mathcal{A} \) gets back is

\[ f(f(m_{11}) \oplus m_{12}) = C_{12} \]
\[ f(f(m_{21}) \oplus m_{22}) = C_{22} \]
Property of PRF

If we find some PRF in table, and if entry found in table then it simply off that entry.

So if

\[ f(m_{11}) \oplus m_{12} = f(m_{21}) \oplus m_{22}. \]

We add 'B' string to \( m_{12} \) & \( m_{22} \).

So if \( f(m_{11}) \oplus m_{12+B} = f(m_{21} + m_{22+B}) \),

So, now we can query for \( m_{12+B} \) & \( m_{22+B} \) separately and get them all.

So we need to see with what probability can we get such a pair any 2 pain match.

So our forgery will now be:

\[ m_{21}, m_{22+B} \rightarrow (C_{21}, C_{22}), C_{23}, \ldots, C_{2n} \]

The 2nd Block \( C_{22} \) will be from the separate query we made for \( m_{11} + (m_{12+B}) \).

So adv of adversary is \( \geq \frac{q(q-1)}{N+1} \)