CSA E0 235: Cryptography	March 29,2015
Lecture 5	
Instructor: Arpita Patra	*Submitted by: Abdullah S*

We earlier saw MAC *(Gen,Mac,Vrfy)* is used to provide integrity i.e. the receiver should be able to verify that the message is from the intended source and no adversary has tampered with the message.Before formally defining security of MAC, we need to consider the power of adversary and what is the break. Adversary can get to see MAC tags for messages of his choice, known as the *Chosen-Message Attack*. Adversary generating a valid message tag pair for a previously unseen message is considered as break of MAC security.

1 Definition of Security for MAC

1.1 CMA-Security

Notion of CMA-Security is captured using the following experiment
MAC-forge$_{A,\Pi}$ (n)= (Gen,Mac, Vrfy) where the

1. Adversary A, is given access to a MAC-oracle in the training phase.Let Q denote the queries that the adversary asks.
2. Adversary outputs a message ,m $\notin Q$.
3. A succeeds if Vrfy $_k$ (m,t)=1.

Definition 1 A message authentication code Π= (Gen,Mac, Vrfy) is existentially unforgeable under an adaptive *chosen-message attack*,or just secure, if for all PPT adversaries A, there is a negligible function negl(.) such that: Pr[*MAC-forge*$_{A,\Pi}$ (n) = 1] $\leq$ negl(n). $\diamondsuit$

1.2 Strong CMA-Security

A stronger notion of CMA-Security is captured using the following modified experiment *MAC-sforge*$_{A,\Pi}$ (n)= (Gen,Mac, Vrfy). It is the same as MAC-forge,except that now the set Q contains pairs of oracle queries and their associated responses $< m, t > \in Q$ .The adversary A succeeds (and experiment Mac-sforge evaluates to 1) if and only if A outputs (m, t) such that Vrfy$_k(m, t) = 1$ and $< m, t > \notin Q$.

Definition 2 A message authentication code Π= (Gen,Mac, Vrfy) is strongly secure, if for all PPT adversaries A, there is a negligible function negl(.) such that
Pr[*MAC-sforge*$_{A,\Pi}$ (n) = 1] $\leq$ negl(n). $\diamondsuit$

2 Construction of Fixed-Length MAC

To construct fixed-length MAC, PRF can be used. Consider a PRF $\{0,1\}^n \times \{0,1\}^{\rightarrow}\{0,1\}^n$. Π =*(Mac, Vrfy)* is a fixed-length MAC constructed as follows

- Mac: On input key k and message $m \in (0,1)^n$, tag t is calculated as t := F_k(m).
- Vrfy: Output 1 if and only if t := F_k(m).

Theorem 1 *If F is a PRF , then* Π *is a CMA-secure MAC.*

Proof Intuition behind the proof is that , if Π was not CMA-secure, then Π can be used to construct a distinguisher D for the PRF.Since if a truly random function f, was used instead of the PRF, adversary would not able to do better than guessing the tag with success probability of atmost 2^{-n}. ■

3 Domain Extension

The above construction is capable generating MAC tags for fixed-length messages.So, we need to use this to construct MAC tags for arbitrary length messages. General idea is to break the large message into blocks of fixed-size and construct MAC tag for each block using a secure fixed length MAC $\Pi' = (Mac', Vrfy')$.

- Compute tag t_i := $F_k(m_i)$ for each block i and output $t1||t2||..t_d$ as the tag. Adversary can reorder the blocks and its corresponding tags to generate a valid $< m, t >$ pair, *block-reordering attack* i.e If tag t=t_1, t_2 is a valid tag on message $m = m_1, m_2$,then $< (m_2, m_1), (t_2, t_1) >$ is a new valid $< m, t >$ pair.
- The above attack can be prevented if each block has some identifier. Add a block id to each block before authenticating each block .Tag is calculated as t_i := $F_k(i||m_i)$.Adversary can still generate a new valid $< m, t >$ pair by dropping some blocks from the end,*truncation attack* i.e If tag t=t_1, t_2, t_3 is a valid tag on message $m = m_1, m_2, m_3$,then $< (m_1, m_2), (t_1, t_2) >$ is a new valid $< m, t >$ pair.
- The above attack can be prevented if receiver is able to know the length of the message. Add block length l, to each block before authenticating. Now tag t is calculated as t_i := $F_k(l||i||m_i)$.Suppose attacker learns two valid $< m, t >$ pairs for messages of same length, if,$< (m1, m2), (t1, t2) >$ *and* $< (m1', m2'), (t1', t2') >$,he can still generate a new valid $< m, t >$ pair by combining blocks of the two messages,*mix and match attack* . Here $< (m1, m2'), (t1, t2') >$ is a new valid $< m, t >$ pair.
- If each message has a unique id then the above attack is also prevented. Add a random identifier r, for every message block before authenticating. Now tag t_i := $F_k(r||l||i||m_i)$. This looks secure for all of the above attacks[security has to be proved] but is highly inefficient. For message of size d blocks, tag will be of size $4d$ and Π' is invoked $4d$ times.

3.1 Construction of MAC

If $\Pi' = (Mac', Vrfy')$ be a fixed-length MAC, then MAC Π =*(Mac, Vrfy)* for arbitrary length is constructed as follows

- Mac: On input key k and message $m \in (0,1)^n$, m is parsed into d blocks each of size n/4. For each block i, tag $t_i := F_k(r||l||i||m_i)$. Output the tag $t=<r,t_1||t2||....t_d>$
- Vrfy: On input t and key k, find tag t'_i using r for each block. Output 1 if and only if $t'_i := t_i$ for all i.

Theorem 2 *If $\Pi' = (Mac', Vrfy')$ is a secure MAC for messages of fixed-length , then $\Pi = (Mac, Vrfy)$ in construction 3.1 is a secure MAC for messages of any arbitrary length.*

Proof Idea behind the proof is that the adversary cannot generate a new message from the learned message blocks and a valid tag cannot be generated by the adversary on a new block as long as Π' is secure. We prove by *contradiction.* Assume Π is not secure. $\exists$ a PPT adversary A, for Π with success probability *>1/p(n).* We use this adversary A to build adversary A' for Pi'.

1. For all queries sent by A in the training phase , A' parses messages into blocks of size n/4 and send $< r||l||i||m_i >$ to its challenger. Sends the valid tags from its challenger back to A.
2. When A sends a $< m, t >$ pair , A' checks if its a new message by checking if there is atleast one unauthenticated block. If a new block is present sends $< m_i, t_i >$ to its challenger . We prove that the $Prob[successof A']$ is the same as A.

Let the number of MAC oracle queries made by A be $q(n)$. Consider *NewBlock* to be the event that A tries to output a valid tag on a block that was previously unauthenticated and *Repeat* be the event that the same random identifier appears in two of the tags returned by the MAC oracle in experiment Mac-forge$_{A,\Pi}$(n) experiment. In case of the *repeat* event happening , adversary can mount a *mix and match* attack and succeed in breaking Π. Thus we need to prove that this probability is negligible. In case of the *repeat* event does not happen,two cases arise ,experiment succeeding given *NewBlock* event occurs and experiment succeeding given *NewBlock* event does not occur($\overline{NewBlock}$). We have to prove that when *NewBlock* event does not occur , probability that A succeeds is zero . This is the crux of the proof.

$$
\begin{aligned}
Prob[\ Mac\text{-}forge_{A,\Pi}(n) = 1] = \ & Pr[\ Mac\text{-}forge_{A,\Pi}(n) = 1 \wedge Repeat] \\
& + Pr[Mac\text{-}forge_{A,\Pi}(n) = 1 \wedge \overline{Repeat} \wedge NewBlock] \\
& + Pr[Mac\text{-}forge_{A,\Pi}(n) = 1 \wedge \overline{Repeat} \wedge \overline{NewBlock}] \\
\leq\ & Pr[\ Mac\text{-}forge_{A,\Pi}(n) = 1 \wedge Repeat] \\
& + Pr[Mac\text{-}forge_{A,\Pi}(n) = 1 \wedge NewBlock] \\
& + Pr[Mac\text{-}forge_{A,\Pi}(n) = 1 \wedge \overline{Repeat} \wedge \overline{NewBlock}]
\end{aligned}
$$

- Consider the first term, *Pr[Repeat]*
  Let the number of MAC oracle queries made by A *is q(n).* The probability of event *Repeat* is exactly the probability that $r_i = r_j$ for some i$\neq$ j where each r_i is chosen uniformly from $2^{n/4}$. Thus $Pr[Repeat] \leq q(n)^2/2^{n/4}$ and is negligible.

- Consider the final term on the *RHS*
  Since repeat event has not occurred , all the queries made during the training phase have unique r. Thus $r_i \neq r_j$ for all $i \neq j$.

  - i)If random identifier is not equal to any of the unique r_i , then it is a new block. Implies we have $r=r_j$ for some query j
  - ii)If $|m| \neq |m_j|$ then length part of the block will be different and it also becomes a new block.
  - iii)$|m| = |m_j|$, atleast one of the blocks must be unauthenticated,say the k^{th} block denoted by m^k is not equal to k^{th} block of the j^{th} query message denoted by m_j^k. But the only to have tag $k = tag_j^k$ is to have $m^k = m_j^k$. But this is also not possible.

  Hence $Pr[Mac\text{-}forge_{A,\Pi}(n) = 1 \wedge \overline{Repeat} \wedge \overline{NewBlock}]=0$

- $Pr[Mac\text{-}forge_{A,\Pi}(n) = 1 \wedge NewBlock]$
  This relies on security of Π'. If *NewBlock* occurs and if $Pr[Mac\text{-}forge_{A,\Pi}(n) = 1\ ]$ then the tag on every block of the message is valid (including tag on the new block).Thus adversary A' outputs this block and its corresponding tag. This means that whenever A succeeds with a *NewBlock* ,A' also succeeds. Thus if A breaks Π with non-negligible probability, A' will break the CMA-security of Π' with non negligible probability,since other two terms of the equation is proved to be non-negligible. This is a contradiction.

■

4 Authenticated Encryption

In the previous chapter, we saw how to obtain secrecy in the private-key setting using encryption. In this chapter, we have shown how to ensure integrity using message authentication codes. Indeed both are necessary security features ,one might want to achieve both goals simultaneously.This leads to Authenticated Encryption .Thus ,for a scheme to be an AE scheme ,we need both secrecy and integrity.
For secrecy, we demand CCA security: no PPT attacker should be able to non-negligibly distinguish between encryption of two messages of its choice, even if it has access to encryption and decryption oracle service.
For integrity/authentication, we demand something similar to CMA security for MAC. No PPT attacker should be able to come up with a valid ciphertext for a new message for which the adversary has not seen a ciphertext before.The *unforgeable* experiment,Enc-Forge $_{A,\Pi}(n)$ is defined as follows :

1. Run Gen(1^n) to obtain a key k.
2. The adversary A is given input 1^n and access to an encryption oracle $Enc_k()$. Let Q denote the set of all queries that A asked its encryption oracle.

3. The adversary outputs a ciphertext c

4. Output of the experiment is 1 if and only if m := Dec_k(c) is a valid message for $\exists m \notin Q$

Definition 3 A private-key encryption scheme Enc-Forge $_{A,\Pi}$(n) is *unforgeable* if for all probabilistic PPT adversaries A, there is a negligible function *negl(.)* such that: *Pr[Enc-Forge $_{A,\Pi}$(n) = 1] ≤ negl(n).* ♢

4.1 Formal definition of Authenticated Encryption

A symmetric-key cipher $\Pi = (Gen, Enc, Dec)$ is an authenticated cipher if both the following holds:

- Π *is CCA-secure*
  For every PPT adversary A participating in the CCA-experiment, there is a negligible function negl1(), such that: *Pr[PrivK$^{cca}_{A,\Pi}$ (n) ≤ 1/2 + negl1(n)]*

- Π *is unforgeable* For every PPT adversary A participating in the *unforgeable encryption* experiment, there is a negligible function negl2(), such that: *Pr[Enc-forge$_{A,\Pi}$ (n) ≤ negl2(n)]*