CSA E0 235: Cryptography February 23rd, 2016

Scribe for Lecture # 10

Instructor: Arpita Patra *Submitted by: Mukesh Makwana*

1 Recall

1. We studied security definitions of MAC - cma, strong cma, cmva and strong cmva.
2. Then we saw construction of PRF.
3. And How to find a tag for long message.

2 Topics to be covered

1. Some rectifications regarding previous class.
2. Authenticated Encryption (AE)
   (a) Definition
   (b) Construction from cpa-secure SKE with scma-secure MAC
   (c) Proof
   (d) AE implies cca-secure SKE

3 MAC

A *Message authentication code* is a special case *message authentication* scheme, where message m is sent with a tag t which authenticates the originality of the fact that message m was actually sent by *Sender* and has not been tempered by the *Adversary* while in the process of transmission.

3.1 cma-security for MAC

Let $\prod$ = (Gen, Mac, Vrfy) and n be the scheme. Then the message authentication experiment $Mac\text{-}forge_{A,\prod}$(n) will be,

1. A random key $k \leftarrow \{0,1\}^n$ is chosen.
2. The adversary A is given oracle access to $Mac_k(.)$ and outputs a pair (m,t). Formally, $(m,t) \leftarrow A^M ac_k(.)(1^n)$. Adversary asks some set of queries lets denote it as Q.
3. The output of experiment is defined to be 1 if and only if $m \notin$ Q and $\text{Vrfy}_k(m,t) = 1$ else 0.

So the definition is; A message authentication code $\prod$ = (Gen, Mac, Vrfy) is existentially unforgeable under an adaptive chosen-message attack, or just secure, if for all probabilistic polynomial-time adversaries A, there exists a negligible function negl such that:

$$\Pr(Mac - forge_{A,\prod}(n) = 1) \leq \text{negl(n)}$$

3.2 strong-cma-security for MAC

Message will be strong cma-secure if in the above experiment instead of just saying m does not belongs to Q, we say message-tag pair (m,t) does not belong to Q.

4 Rectifications

1. Its need *NOT* to be true to need randomized MAC to satisfy scma-security. This is basically because even if we use deterministic approach in scma-secure scheme, we will get same tag for a message every time and verifier won't output 1 if he finds that message-tag pair (m,t) from Q.

2. Canonical verification is simply recomputing the tag for the message and check whether tag in pair (m,t) produced by adversary is same as recomputed tag. That means algorithm should be deterministic in nature. And from above point it need not to be randomized to satisfy scma security. So its clear that MAC that is cma secure will always be scma secure with canonical verification.

3. Every deterministic MAC has canonical verification. Rather then decrypting, we can just check whether the tag t given for a message m is same as recomputed tag for verification process with deterministic MAC because for a message m it will produce same tag t every time.

4. PRF-based scheme to construct a MAC is thus scma-secure if it is deterministic and provably cma-secure.

5 Authenticated Encryption (AE)

Why? We have studied how to obtain *secrecy* and *integrity*. But what if we want to achieve both simultaneously? It would be the best practice to have authenticated encryption by default which fulfills both secrecy and integrity in private-key setting.

5.1 Definition:

Here our notion of secrecy will be *cca* secure because it is more powerful than *cpa-security* and *coa-security*, as we know *adversary* has Encryption and Decryption oracle. For integrity it will be unforgeable under adaptive cma-secure. So $\prod = (Gen, Enc, Dec)$ is authenticated encryption if, $\prod = (Gen, Enc, Dec)$ is cpa-secure and, $\prod$ also provides ciphertext integrity, which implies that its hard to come up with a ciphertext that has a meaningful decryption even after sufficient training.

5.2 The Ciphertext Integrity experiment

Let $\prod$ = (Gen, Enc, Dec) be the scheme. Then the Ciphertext Integrity experiment $CiIn_{A,\prod}$(n) will be,

1. Verifier obtains a k(Key) using $Gen(1^n)$ algorithm.
2. *Adversary* is given access to Encryption Oracle $Enc_k()$, so he asks t number of queries (Let we call set of all queries to be Q) (that is Training phase for Adversary). And in Challenge phase outputs a ciphertext c to Verifier.
3. If Verifier using $Dec_k()$ finds that $m \neq \perp$ and $c \notin Q$ then outputs 1, else 0.

A private-key encryption $\prod$ if for all probabilistic polynomial-time adversaries A, there is a negligible function `negl` such that:

$$\Pr(CiIn_{A,\prod}(n) = 1) \leq \text{negl(n)}.$$

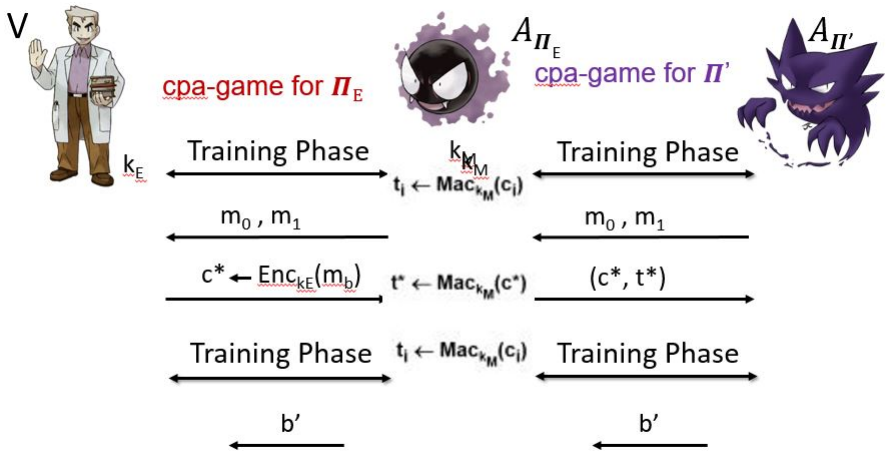
5.3 Construction of Encrypt-then-authenticate:

We have already seen in previous lecture that Encrypt-and-authenticate fails because secure MAC does not provide any secrecy. And Authenticate-then-encrypt fails for CBC-mode of encryption with MAC. So we are left with Encrypt-then-authenticate which is also secure. Lets again have a look at its construction:

First we define private-key encryption scheme as $\prod_E = (Enc, Dec)$ and message authentication code as $\prod_M = (Mac, Vrfy)$ with a private-key encryption scheme $\prod$' = (Gen', Enc', Dec') as follows:

1. Gen': Choose a uniform n-bit key i.e. $k_E, k_M \in \{0,1\}^n$ and output them.
2. Enc': Using key pair (k_E, k_M) and message m, compute $c \leftarrow Enc_{k_E}(m)$ and $t \leftarrow Mac_{k_M}$(c). Output the ciphertext (c,t).
3. Dec': Using key pair (k_E, k_M) and ciphertext (c,t), first we need to check that the $Vrfy_{k_M}$(c,t) = 1, if *yes* then we need to further check for output of $Dec_{k_E}(c)$ else just output $\perp$.

***Lemma #1**: If $\prod_E$ is cpa-secure then we say $\prod$' is cpa-secure.*
-proof-
We go by contradiction, assuming $\prod_E$ to be a cpa-secure but $\prod'$ not. So what we are saying in $\prod'$ scheme, there is an *Adversary*, lets call it $A_{\prod'}$ who wins with a non-negligible probability and using this adversary we need to break the $\prod_E$ scheme.
So what happens is $A_{\prod'}$ sends messages to $A_{\prod_E}$, which simply forwards them to Verifier and in return gets a ciphertext c_i. $A_{\prod_E}$ computes a key k_M and generates a tag on those ciphertext using algorithm $Mac_{k_M}(c_i)$ and returns back to $A_{\prod'}$ (c_i, t_i) (that is Training Phase). In Challenge phase $A_{\prod'}$ sends two messages m_0 and m_1 of same length to $A_{\prod_E}$ which again forwards them to Verifier and gets a $c* \leftarrow Enc_{k_E}(m_b)$ where b could 0 or 1. Again $A_{\prod_E}$ generates a tag on c* and forwards (c*, t*).

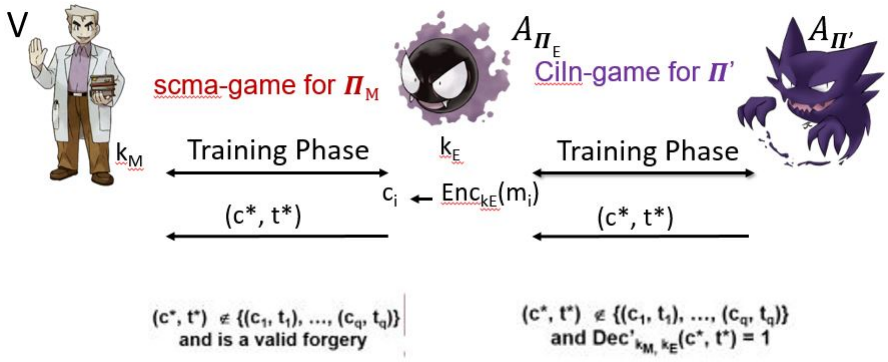


Now, we say if $A_{\prod'}$ has a non-negligible advantage to win this game so does $A_{\prod_E}$. Which states $\prod_E$ cannot be cpa-secure. But we already assumed it be cpa-secure, which contradicts our this assumption. And we say $A_{\prod'}$ is also cpa-secure.

***Lemma**: If $\prod_M$ is scma-secure then $\prod'$ is scma-secure.*

-proof-

Same as we did in earlier proof we are going to assume that $\prod_M$ is scma secure but $\prod'$ is not. So there is an adversary lets call it $A_{\prod'}$ with non-negligible probability can break the $ciphertext - integrity$ of $\prod'$ and using that adversary we need to break scma-security of $\prod_M$. Similar to our Ciphertext-Intigrity game here $A_{\prod'}$ will send some messages to $A_{\prod_M}$, who will generate a key k_E and encrypt the message with that key using $Enc_{K_E}(m_i)$ and forward it to Verifier. Verifier in return will generate a valid tag t_i on ciphertext c_i and send back the (c_i, t_i) pair to $A_{\prod_M}$ who will simply forward it to $A_{\prod'}$ (that is training-phase). In challange-phase $A_{\prod'}$ will come up with a (c*, t*) will be sent to verifier by $A_{\prod_E}$. If its a valid (c*, t*) pair for $\prod_M$ then we say $A_{\prod_M}$ also has non-negligible advantage to win as $A_{\prod'}$ and thus $\prod_M$ is not scma-secure. But we assumed that $\prod_M$ is scma-secured which contradicts. Thus we say that even $\prod'$ is scma-secure.



But the question now arises is, does this kind of reduction proof stands for Authenticated Encryption or *Authenticate then Encrypt*. As in earlier reduction proof, Aversary $A_{\prod'}$ may not come up with a valid c* which contains a valid message and tag pair (m, t) or could be the case c* belongs to set of all messages from training phase. So reduction proof same as of *lemma* #2 wont work here.

5.4 AE implies cca-security

AE and cca-secure schemes differ in only the term of integrity which is provided by AE and not by cca-secure. So AE is more stronger than cca-secure. ***Theorem**: Every Authenticated Encryption is cca-secure.*

-proof-

Using contradiction, i.e. $\prod$ is not a cca-secure.

$$Pr(Priv^{cca}_{A,\prod}(n) = 1) > \frac{1}{2} + \frac{1}{p(n)} \tag{1}$$

where, p(n) stands for polynomial.

Now, Adversary can win this game in two different situation which is based on type of query. i.e.

1. Valid query or Non-trival query (NTQ).
2. Not a Valid query or not a Non-Trivial query($\overline{NTQ}$).

A NTQ is, if ciphertext c did not come from set of queries to Encryption oracle or challenge chiphertext and it decrypts to a valid message. And $\overline{NTQ}$ is which decrypts to invalid message. So adversary can go for either of it, i.e.

$$Pr\left(Priv^{cca}_{A,\prod}(n) = 1\right) = Pr\left(Priv^{cca}_{A,\prod}(n) = 1 \wedge NTQ\right) + Pr\left(Priv^{cca}_{A,\prod}(n) = 1 \wedge \overline{NTQ}\right) \tag{2}$$

$$\frac{1}{2} + \frac{1}{p(n)} < Pr\left(Priv^{cca}_{A,\prod}(n) = 1 \wedge NTQ\right) + Pr\left(Priv^{cca}_{A,\prod}(n) = 1 \wedge \overline{NTQ}\right) \tag{3}$$

$$\frac{1}{2} + \frac{1}{p(n)} < Pr\left(NTQ\right) + Pr\left(Priv^{cca}_{A,\prod}(n) = 1 \wedge \overline{NTQ}\right) \tag{4}$$

So now there are two cases, as:

Case #1: Pr($Priv^{cca}_{A,\prod}(n) = 1 \wedge \overline{NTQ}) > \frac{1}{b(x)}$; *where b(x) is a polynomial*

Its just that Adversary can win with non-negligible advantage and without NTQ. It implies that he can win cpa-security. Thus the scheme can not be AE-secure.

Case #2: Pr(NTQ) $> \frac{1}{a(x)}$; *where a(n) is a polynomial*

There are two games played, one between A_{CiIn} and $A_{\prod}$ is cca-ind-game, and another between A_{CiIn} and Verifier is CiIn-game. All queries of $A_{\prod}$ are simply forwarded to verifier by A_{CiIn}. As $A_{\prod}$ has decryption oracle and so to play along A_{CiIn} chooses a random i and till that ith numbered query is made he sends back respective message if c was queried

earlier to encrytion oracle else if it would have been the challenge ciphertext itself then send a invalid message ($\perp$). But at *ith* query, c is forwarded to Verifier, where if c is the first NTQ then A_{CiIn} then cca-ind game with $A_{\prod}$ is correct. We say that probability of A_{CiIn} to win the CiIn-game is same as first NTQ is made. i.e.

$$Pr\left(CiIn_{A_{CiIn}}(n)=1\right)=Pr\left(NTQ\wedge I\right) \tag{5}$$

$$Pr\left(CiIn_{A_{CiIn}}(n)=1\right)=Pr\left(NTQ\right)Pr\left(I\right) \tag{6}$$

$$Pr\left(CiIn_{A_{CiIn}}(n)=1\right)>\frac{1}{a(x)}.\frac{1}{b(x)} \tag{7}$$

$$Pr\left(CiIn_{A_{CiIn}}(n)=1\right)>non-negl(n) \tag{8}$$

(Above I is the event when ith query is made.)

References

1. Arpita Patra. http://drona.csa.iisc.ernet.in/arpita/Cryptography16.html.

2. Katz, Jonathan, and Yehuda Lindell. Introduction to Modern Cryptography. CRC Press, 2014.