Authenticated Encryption
+
Message Authentication Codes

CS 284, Lecture 12, Autumn 2021
Outline

1. Chosen-Ciphertext Security for Encryption
2. Message Authentication Codes (MACs)
3. MAC Constructions from a Blockcipher
4. Combining Encryption with a MAC
Outline

1. Chosen-Ciphertext Security for Encryption
2. Message Authentication Codes (MACs)
3. MAC Constructions from a Blockcipher
4. Combining Encryption with a MAC
A typical attack setting: Server connected to the Internet

```
Server
k
m ← Dec(k, c)
If check-format(m):
    Send next msg
Else:
    Send Error
```
A typical attack setting: Server connected to the Internet

\[ \text{Server} \]
\[ k \]
\[ m \leftarrow \text{Dec}(k, c) \]

If check-format \( m \):
Send next msg
Else:
Send "Error"

\[ m \leftarrow \text{Dec}(k, c) \]

→ "Chosen-Ciphertext Attacks" (CCA) vs CPA
Chosen-Ciphertext Security: Plan for Definition

Goal: Definition for hiding all plaintext info against such attackers
→ Follow left/right idea from CPA.
→ Need to allow A ability to inject ciphertexts and observe responses.

* Like CPA, but give A a $\text{Dec}(k_1)$ oracle!
CCA Definition

Def Let \( T = (\text{Enc}, \text{Dec}) \) be a randomized encryption scheme with respect to \( K, M, R, C \). Assume \( M \subseteq \{0,1\}^* \). Let \( A \) be an algorithm. Define

\[
\text{Expt}_{\text{CCA}}^T(A) = \begin{cases} \Pr[K \in \text{K}, b \in \{0,1\} : \text{Run } A^{L_{\text{Rand}}, \text{Dec}(K, \cdot)} \quad \text{A halts with output } b \} \\
\text{If } A \text{ ever queried } \text{Dec}(K, \cdot) \text{ on some } c \text{ previously output by } L_{\text{Rand}}: \text{ output } 0 \\
\text{If } b = 5 \text{ output } 1 \text{ Else output } 0 \end{cases}
\]

Oracle \( L_{\text{Rand}}(m_0, m_1) \)

- If \( m_0, m_1 \) are not same length: return 1
- Pick \( r \in \mathbb{R} \)
- \( c = \text{Enc}(K, m_0, r) \)
- Return \( c \)

The CCA advantage of \( A \) against \( T \) is

\[
\text{Adv}_{\text{CCA}}^T(A) = \left| \Pr[\text{Expt}_{\text{CCA}}^T(A) = 1] - \frac{1}{2} \right|
\]
CCA Example

Let $T=(\text{Enc}, \text{Dec})$ be defined by $\text{Enc}(k,m,r) = (r, \text{Elk}(r) \oplus m)$, $\text{Dec}(k_1(r,r)) = \text{Elk}(r) \oplus y \rightarrow \oplus$, $b=1$.

Claim There is an efficient A such that $\text{Adv}_{\text{CCA}}(\lambda) = 1/2$.

A

Fix $m, m_1 \epsilon M$, distinct, $m_0 \not= \emptyset$

Query $(r, y) \leftarrow O_1(m, m_1)$

Set $y' = y \oplus m_0$.

Query $m' \leftarrow O_2((r, y'))$ // $\text{Dec}(k_1(r,y'))$

If $m' = \emptyset$ output 0, Else output 1.
Claim: \( b = 0 \implies \text{Dec}(k_1(r, y)) = 0^l \)

Claim: \( b = 1 \implies \text{Dec}(k_1(r, y)) \neq 0^l \)
CCA Security $\simeq$ Security Against "Malleability"

Plan: Make it "hard" for $A$ to come up with any $c' \neq c$ without getting caught!
Outline

1. Chosen-Ciphertext Security for Encryption
2. Message Authentication Codes (MACs)
3. MAC constructions from a Blockcipher
4. Combining Encryption with a MAC
Message Authenticity

\[
\text{Sender} \xrightarrow{m} \text{Receiver} \xrightarrow{\text{\textasciitilde}} \begin{array}{c}
\text{\textasciitilde} \text{really sent} \\
\text{by sender?}
\end{array}
\]

- Different from secrecy? Not trying to hide \( m \) from 
- Can ask for both (will later)
Encryption May not Provide Message Authenticity

- May be able to get receiver to accept another \( \hat{m} \) undetected

  - ex: Enc is OTP, then flipping bits of \( c \) will work
New tool for Authentication: Message Authentication Codes (MAC)

Will use a function $\text{Mac} : \{0,1\}^n \times M \rightarrow \tau$ — “tags”

$\tau \leftarrow \text{Mac}(k \cdot m)$

$\hat{m}, \hat{t} \leftarrow \&$

$\tau = \text{Mac}(k \cdot \hat{m})$

*Goal:* Should be hard for $\bullet$ to find correct $\hat{t}$ for some $\hat{m}$

*Give up on “replay attacks”, where $\bullet$ forwards some $m, t$ that really did come from sender.*
MAC Security: Ideas

Will use a function $\text{MAC} : 12 \times M \to T$

1. Tries to "force" a $\hat{t}$ on same $\hat{m}$
2. Sets to see many forged $t$ on message of its choice.
3. Sets to see if receiver will accept many $\hat{m}, \hat{t}$ inputs, again of its choice.
MAC Security Definition

Let $\text{Mac}: K \times M \rightarrow T$ and let $A$ be an adversary. Define

$$\text{Exp}^{\text{uf}}_{\text{Mac}}(A)$$

Pick $k \leftarrow K$

Run $A$ on $\text{Mac}(k, \cdot), \text{Vrfy}_k(\cdot, \cdot)$ until it halts

If $A$ ever queried 2nd oracle an $\hat{\omega}, \hat{\xi}$ such that

1. $\hat{\xi} = \text{Mac}(k, \hat{\omega})$, and
2. $\hat{\omega}$ was never previously queried to $\text{Mac}(k, \cdot)$ oracle

then output 1.

Else output 0.

Define $\text{Adv}^{\text{uf}}_{\text{Mac}}(A) = \Pr[\text{Exp}^{\text{uf}}_{\text{Mac}}(A) = 1]$. 

Oracle $\text{Vrfy}_k(\hat{\omega}, \hat{\xi})$

If $\text{Mac}_k(\hat{\omega}) = \hat{\xi}$: return 1
Else: return 0
To win: Send a correct message/tag pair to Vrty with asking for a tag.
MAC Example

Define $Mac(k, m) = k \oplus m$.

- $Mac(k, \cdot)$, $Vrfy_k(\cdot, \cdot)$
- $\hat{t} \in Mac(k, \emptyset)$
- $t' \in \hat{t} \oplus 1^l$ "w'
- Query $Vrfy_k(1^l, t')$

- never queried $m'$ to $Mac(k, \cdot)$
- $Mac(k, 1^l) = k \oplus 1^l = \hat{t} \oplus 1^l = t'$

$Mac_k() \xrightarrow{\hat{t}} A$

$\hat{w} = \emptyset \rightarrow \hat{t} = k$ (!)

$m' = 1^l$
$t' = \hat{t} \oplus 1^l$

$\uparrow$ forgery
MAC Example

Define $Mac(k, m) = k \oplus m$.

Query $\emptyset$ to $Mac(k, \cdot)$. Call result $t$.
Set $\hat{m} = 1^l$, $\hat{t} = t \oplus 1^l$.
Query $(\hat{m}, \hat{t})$ to $Vrfy(k, \cdot)$.

$A\hat{t} \not\in Mac(k, \hat{m})$ since:

$Mac(k, \hat{m}) = k \oplus 1^l = \emptyset \oplus 1^l$ 
$(\emptyset = k \oplus \emptyset = \emptyset)$.
and $\hat{m} = 1^l$ was never queried to $Mac$. 

\[
\begin{align*}
\hat{m} & = 1^l \\
\hat{t} & = t \oplus 1^l
\end{align*}
\]
Outline

1. Chosen-Ciphertext Security for Encryption
2. Message Authentication Codes (MACs)
3. MAC constructions from a Blockcipher
4. Combining Encryption with a MAC
Constructing MACs

1. In some sense, "A PRP is also a good MAC, as long as its output size is not too small."

2. AES is a good PRP, therefore also a good MAC (128-bit output is enough). But it only takes 128-bit inputs.

⇒ So we build MACs for larger messages from AES.
MACs for Longer Messages

Naive attempt for two blocks:

\[ F(k_1, m_1 || m_2) = \text{AES}(k, m_1) \parallel \text{AES}(k, m_2) \]

\[ t_1 \parallel t_2 \]

\[ t_2 \parallel t_1 \]

is tag for \( m_2 || m_1 \)
MACs for Longer Messages

Naive attempt for two blocks:

\[ F(k_1, m_1 || m_2) = \text{AES}(k_1, m_1) || \text{AES}(k_1, m_2) \]

Attack \( A \) with \( \text{Adv}^F_\rho(A) = 1 \):

1. Query \( 0^{256} || 1^{256} \) to \( \text{Mac}(k, \cdot) \). Get \( t \).
2. Parse \( t \) as \( t_0 || t_1 \).
3. Query \( \hat{m} = 0^{128} || 0^{128}, \hat{t} = t_0 || t_0 \) to \( \text{Vrfy}(k, \cdot) \).
Another Example

\[ F(k, m_1 || m_2) = AES(k, m_1) \oplus AES(k, m_2) \]

\[ m_1 = m_2, \quad t = 2^{128} \quad \text{works!} \]
A Practical MAC: CBB - MAC

Key $k \in \{0,1\}_{128}$

1. Let $\text{len} \in \mathbb{Z}_{128}$ be the length of $m$, encoded as a 128-bit block.
2. $\overline{m} \leftarrow \text{pad}_{128}(m) \parallel \text{pad}$ to multiple of 128
3. Parse $\overline{m}$ into blocks $\overline{m}[0], \ldots, \overline{m}[t]$
CBL-MAC without length?

Insecure! Fun exercise.

(Hint: Query $\text{Mac}(k, 0^{127})$ set $t$

Now find a two-block message that will verify with one $t$.}
CBC-MAC with length at end?

Insecure! A bit harder, but still simple.
Outline

1. Chosen-Ciphertext Security for Encryption
2. Message Authentication Codes (MACs)
3. MAC Constructions from a Blockcipher
4. Combining Encryption with a MAC
Plan For Achieving CCA Security

1. Build a CPA-secure encryption scheme \((\text{Enc}, \text{Dec})\)

2. Build a secure MAC \(\text{Mac}\)

3. Build encryption scheme \((\text{Enc}', \text{Dec}')\) that runs both \((\text{Enc}, \text{Dec}) + \text{Mac}\).
Combining CPA-Secure Encryption and a MAC: "Encrypt-then-Mac"

Given $T\mathcal{T} = (\text{Enc}, \text{Dec})$ and Mac, build new encryption scheme $T\mathcal{T}' = (\text{Enc}', \text{Dec}')$:

- $\text{Enc}'(k_1, k_2, m, r) \rightarrow \text{Enc} \rightarrow \text{Mac} \rightarrow \text{output} C = (Y, t)$
- $\text{Dec}'(k_1, k_2, (D, t)) \rightarrow \text{Mac} \rightarrow \text{Dec} \rightarrow m$

If $t$ correct, output $m$

Thus (CS 389): If $T\mathcal{T}$ is "CPA secure" and Mac is "secure" then $T\mathcal{T}'$ is "CCA secure".