Authenticated Encryption

Jeremy, Paul, Ken, and Mike
Objectives

- Examine three methods of authenticated encryption and determine the best solution considering performance and security
Basic Components

Message Authentication Code

+ 

Symmetric Encryption

Both of these components are used as black boxes
Generic Composition

- Note:
  - We separate the tagging and verification algorithm

\[ SE \] - Symmetric encryption scheme
\[ E \] - encryption algorithm
\[ D \] - Decryption Algorithm
\[ MA \] - Message authentication scheme
\[ T \] - tagging algorithm
\[ V \] - tag verifying algorithm
\[ K \] - randomized key generation algorithm
\[ \kappa \] - security parameter, length of the key
\[ k \] - the key
Basic Components

Message Authentication Code (MAC)
- Integrity / Authenticity
  - Integrity of Plaintext (INT-PTXT)
  - Integrity of Ciphertext (INT-CTXT)

Symmetric Encryption
- Privacy
  - Indistinguishability
    - Chosen-plaintext attack (IND-CPA)
    - Chosen-ciphertext attack (IND-CCA)
  - Non-malleability
    - Chosen-plaintext attack (NM-CPA)
    - Chosen-ciphertext attack (NM-CPA)
Integrity

- Integrity of Plaintext (INT-PTXT)
  - Computationally infeasible to produce a ciphertext decrypting to a message which the sender has never encrypted

- Integrity of Ciphertext (INT-CTXT)
  - Computationally infeasible to produce a ciphertext not previously produced by the sender, regardless of whether or not the underlying plaintext is “new”
Integrity of symmetric encryption schemed

\[ SE = (E, K, D) \]

Algorithm \( D^*_K(C) \)
- If \( D_K(C) \neq \bot \), then return 1
- Else return 0

Verification algorithm or Verification oracle

\( E \) – Encryption Algorithm
\( K \)– Randomized key generation algorithm
\( D \) – Decryption Algorithm
Integrity of Authenticated encryption scheme

- The scheme SE is said to be INT-PTXT if the function $\text{Adv}^{\text{int-ptyxt}}_{SE,A_{\text{ptyxt}}}(\cdot)$ (the advantage of $A_{\text{ptyxt}}$) is very small for any adversary whose time-complexity is polynomial in $k$.

- Likewise, the scheme SE is said to be INT-CTXT if the function $\text{Adv}^{\text{int-ctxxt}}_{SE,A_{\text{ctxxt}}}(\cdot)$ (the advantage of $A_{\text{ctxxt}}$) is very small for any adversary whose time-complexity is polynomial in $k$. 
Integrity of Authenticated encryption scheme

Experiment $\text{Exp}_{SE,A_{txt}}^{\text{int-ptxt}}(k)$

$K \leftarrow R \rightarrow K(\kappa)$

If $A_{txt}^{E(\cdot),D^*(\cdot)}(k)$ makes a query $C$ to the oracle $D^*_X(\cdot)$ such that

- $D^*_X(C)$ returns 1, and
- $M \triangleq D^*_X(C)$ was never a query to $E_K(\cdot)$

then return 1 else return 0

Experiment $\text{Exp}_{SE,A_{txt}}^{\text{int-ctxt}}(k)$

$K \leftarrow R \rightarrow K(\kappa)$

If $A_{ctxt}^{E(\cdot),D^*(\cdot)}(k)$ makes a query $C$ to the oracle $D^*_X(\cdot)$ such that

- $D^*_X(C)$ returns 1, and
- $C$ was never a response to $E_K(\cdot)$

then return 1 else return 0

$\text{Adv}_{SE,A_{txt}}^{\text{int-ptxt}}(k) = \Pr[\text{Exp}_{SE,A_{txt}}^{\text{int-ptxt}}(k) = 1]$ \quad \text{Advantages of the adversaries}$

$\text{Adv}_{SE,A_{ctxt}}^{\text{int-ctxt}}(k) = \Pr[\text{Exp}_{SE,A_{ctxt}}^{\text{int-ctxt}}(k) = 1]$

$\text{Adv}_{SE}^{\text{int-ptxt}}(k,t,q_e,q_d,\mu_e,\mu_d) = \max_{A_{txt}} \{ \text{Adv}_{SE,A_{txt}}^{\text{int-ptxt}}(k) \}$ \quad \text{Advantages of the scheme}$

$\text{Adv}_{SE}^{\text{int-ctxt}}(k,t,q_e,q_d,\mu_e,\mu_d) = \max_{A_{ctxt}} \{ \text{Adv}_{SE,A_{ctxt}}^{\text{int-ctxt}}(k) \}$
Indistinguishability

- Indistinguishability of Chosen Plaintext Attack (IND-CPA)

- Indistinguishability of Chosen Ciphertext Attack (IND-CCA)

- If $M_0$ and $M_1$ are encrypted, a ‘reasonable’ adversary should not be able to determine which message is sent.
Σ_κ(LR(.,.,b)), where b \{0, 1\}, to take input (M_0, M_1) \ |M_0| = |M_1|

if b = 0
   C \leftarrow Σ_κ(M_0)
   return C

else
   C \leftarrow Σ_κ(M_1)
   return C

- As was mentioned from Adam’s lecture, we consider the encryption scheme to be “good” if a “reasonable” adversary cannot obtain “significant” advantage in distinguishing the cases b = 0 and b = 1 given access to the left-or-right oracle.
Non-malleability

- Prevents the generation of a ciphertext whose plaintexts are meaningful
- Requires that an attacker given a challenge ciphertext be unable to modify it into another, different ciphertext in such a way that the plaintexts underlying the two ciphertexts are “meaningful related” to each other.
- i.e.
  - Ptxt1: send a check of $100.00
  - Ptxt2: send a check of $1000.00
Non-malleability - Formally

Experiment $\text{Exp}_{SE, A_{cpa}}^{nm-\text{cpa}-b}(b)$

$k \leftarrow R \quad K(\kappa)$
$(\vec{c}, s) \leftarrow A_{cpa}^{E_k(LR(\ldots,b))}(k)$
$\vec{p} \leftarrow D_k(\vec{c})$
$x \leftarrow A_{cpa_2}(\vec{p}, \vec{c}, s)$
return $x$

Experiment $\text{Exp}_{SE, A_{cca}}^{nm-\text{cca}-b}(b)$

$k \leftarrow R \quad K(\kappa)$
$(\vec{c}, s) \leftarrow A_{cca_1}^{E_k(LR(\ldots,b))}(k)$
$\vec{p} \leftarrow D_k(\vec{c})$
$x \leftarrow A_{cca_2}(\vec{p}, \vec{c}, s)$
return $x$

$SE = (K, E, D)$
$b \in \{0, 1\}$
$\kappa \in \mathbb{N}$
$A_{cpa} = (A_{cpa_1}, A_{cpa_2}), 1\ \text{oracle}$
$A_{cca} = (A_{cca_1}, A_{cca_2}), 2\ \text{oracles}$

$Adv_{SE, A_{cpa}}^{nm-\text{cpa}}(k) = \text{Pr}[\text{Exp}_{SE, A_{cpa}}^{nm-\text{cpa}-1}(k) = 1] - \text{Pr}[\text{Exp}_{SE, A_{cpa}}^{nm-\text{cpa}-0}(k) = 1]$

$Adv_{SE, A_{cca}}^{nm-\text{cpa}}(k) = \text{Pr}[\text{Exp}_{SE, A_{cca}}^{nm-\text{cpa}-1}(k) = 1] - \text{Pr}[\text{Exp}_{SE, A_{cca}}^{nm-\text{cpa}-0}(k) = 1]$

$Adv_{SE}^{nm-\text{cpa}}(k, t, q_e, \mu_e) = \max_{A_{cpa}} \{Adv_{SE, A_{cpa}}^{nm-\text{cpa}}(k)\}$ \quad If negligible, NM-CPA Secure

$Adv_{SE}^{nm-\text{cca}}(k, t, q_e, \mu_e) = \max_{A_{cca}} \{Adv_{SE, A_{cca}}^{nm-\text{cca}}(k)\}$ \quad If negligible, NM-CCA Secure
Unforgeability

- Weak Unforgeability against Chosen Message Attacks (WUF-CMA)
  - Adversary $F$ can’t create a new message and tag

- Strong Unforgeability against Chosen Message Attacks (SUF-CMA)
  - Adversary $F$ can’t create a new tag for an existing message
Difficulties

- The notions of authenticity are by themselves quite disjoint from the notions of privacy
  - i.e. Sending the message in the clear with an accompanying (strong) MAC achieves INT-CTXT but no kind of privacy
Relations among notions of symmetric encryption

INT-CTXT \^ IND-CPA \rightarrow IND-CCA \rightarrow NM-CPA

INT-PTXT \^ IND-CPA \rightarrow IND-CPA \rightarrow NM-CCA
Relations among notions of symmetric encryption
Theorem 3.1

\[ \text{INT} - \text{CTXT} \rightarrow \text{INT} - \text{PTXT} \]

\[
Adv_{\text{SE}}^{\text{int-PTXT}}(k, t, q_e, q_d, \mu_e, \mu_d) \leq Adv_{\text{SE}}^{\text{int-CTXT}}(k, t, q_e, q_d, \mu_e, \mu_d)
\]

- A – adversary mounting an attack against integrity of plaintexts of SE
- A’ – adversary mounting an attack against integrity of ciphertexts of SE
- A’ = A

Adversary A'(k)

return A(C)

C - is the winning query

It is initiative that if an adversary violates integrity of plaintexts of a scheme SE = (K,E,D) also violates integrity of ciphertexts of the same scheme
Proposition 3.3

- IND-CCA $\leftrightarrow$ INT-PTXT
- Given a symmetric encryption scheme $SE$ which is IND-CCA secure, we can construct a symmetric encryption scheme $SE$ which is also IND-CCA secure but is not INT-PTXT secure.
Let $SE = (K, E, D)$

We define a $\overline{SE}$ such that $\overline{SE}$ is IND-CCA secure but is not INT-PTXT secure.

Basically a certain known string (or strings) will be viewed by $D$ as valid and decrypted to certain known messages, so that forgery is easy.

However these ‘ciphertexts’ will never be produced by the encryption algorithm, so privacy will not be affected.

$SE = (K, E, D)$

Algorithm $\overline{E}_k (M)$

$C' \leftarrow E_k (M)$

$C \leftarrow 0 || C'$

Return $C$

Algorithm $\overline{D}_k (C)$

Parse $C$ as $b || C'$ where $b$ is a bit $\leftarrow E_k (M)$

if $b = 0$ then $M \leftarrow D_k (C')$; return $M$

Else return 0
IND-CCA $\rightarrow$ INT-PTXT

**Attack**

- Query 10 is a valid ciphertext
- It decrypts to a msg (0) that the adversary never queried of its oracle

Adversary $A^{E_k(\cdot), D_k(\cdot)}(k)$

Submit query 10 to oracle $D_k^*(\cdot)$

$\cdots$

$D_k^*(10) = 0$

$10 \rightarrow 1010$

(little Endian, LSB 1st)

$Adv_{SE,A}^{\text{int-ptxt}}(k) = 1$

A makes zero queries to $E_k(\cdot)$ and one query to $D_k(\cdot)$ totaling 2 bits, and is Certainly poly(k)-time
To prove that $SE$ is IND-CCA secure, it suffices (enough) to associate with any poly(k)-time adversary $B$ attacking $SE$ in the IND-CCA sense such that

$$Adv^{ind-cca}_{SE,A}(k) \leq Adv^{ind-cca}_{SE,B}(k)$$

Adversary $B^{E_k(LR(\cdot,\cdot),D_k(\cdot))}(k)$

for $i = 1,\ldots,q_c + q_d$ do

when $A$ makes a query $M_{i,0}, M_{i,1}$ to its left - or - right encryption oracle do

$A \leftarrow 0 \parallel E_k(LR(M_{i,0}, M_{i,1}, b))$

when $A$ makes a query $C_i$ to its decryption oracle do

Parse $C$ as $b_i \parallel C'_i$ where $b_i$ is a bit

if $b = 0$ then $A \leftarrow D_k(C'_i)$

Else $A \leftarrow 0$

It is easy for $B$ to break the scheme if $A$ can
Other Relations

- Theorem 3.2
  - INT-CTXT $\land$ IND-CPA $\rightarrow$ IND-CCA

- Proposition 3.4
  - INT-PTXT $\land$ IND-CPA (does not) $\rightarrow$ NM-CPA
Security of the Composite Schemes

- **Secure**
  Proven to meet the security requirement, assuming component encryption scheme meets IND-CPA and message authentication scheme is unforgeable under CMA

- **Insecure**
  Some IND-CPA secure symmetric encryption and some message authentication scheme unforgeable under CMA exist that doesn’t meet the security requirement
Generic Composition

Using both functions as black boxes

MAC

Symmetric Encryption
Encrypt-and-MAC

\[ C = \text{Encrypt} (M) \parallel \text{MAC} (M) \]
## Encrypt-and-MAC Security

<table>
<thead>
<tr>
<th>Security</th>
<th>Weak MAC</th>
<th>Strong MAC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Privacy</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IND-CPA</td>
<td>Insecure</td>
<td>Insecure</td>
</tr>
<tr>
<td>IND-CCA</td>
<td>Insecure</td>
<td>Insecure</td>
</tr>
<tr>
<td>NM-CPA</td>
<td>Insecure</td>
<td>Insecure</td>
</tr>
<tr>
<td>Integrity</td>
<td></td>
<td></td>
</tr>
<tr>
<td>INT-PTXT</td>
<td>Secure</td>
<td>Secure</td>
</tr>
<tr>
<td>INT-CTXT</td>
<td>Insecure</td>
<td>Insecure</td>
</tr>
</tbody>
</table>
MAC-then-Encrypt

\[ C = \text{Encrypt}(M \ || \ \text{MAC}(M)) \]
# MAC-then-Encrypt Security

<table>
<thead>
<tr>
<th>Security</th>
<th>Weak MAC</th>
<th>Strong MAC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Privacy</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IND-CPA</td>
<td>Secure</td>
<td>Secure</td>
</tr>
<tr>
<td>IND-CCA</td>
<td>Insecure</td>
<td>Insecure</td>
</tr>
<tr>
<td>NM-CPA</td>
<td>Insecure</td>
<td>Insecure</td>
</tr>
<tr>
<td>Integrity</td>
<td></td>
<td></td>
</tr>
<tr>
<td>INT-PTXT</td>
<td>Secure</td>
<td>Secure</td>
</tr>
<tr>
<td>INT-CTXT</td>
<td>Insecure</td>
<td>Insecure</td>
</tr>
</tbody>
</table>
Encrypt-then-MAC

\[ C = \text{Encrypt (M)} \| \text{MAC (Encrypt (M))} \]

Algorithm $\overline{K}(k)$
- $K_e \leftarrow K_e(k)$
- $K_m \leftarrow K_m(k)$
- Return $\langle K_e, K_m \rangle$

Algorithm $\overline{E}_{\langle K_e, K_m \rangle}(M)$
- $C' \leftarrow E_{K_e}(M)$
- $\tau' \leftarrow T_{K_m}(C')$
- $C \leftarrow C' \| \tau'$
- Return $C$

Algorithm $\overline{D}_{\langle K_e, K_m \rangle}(C)$
- Parse $C$ as $C'' \| \tau''$
- $M \leftarrow D_{K_e}(C'')$
- $v \leftarrow V_{K_m}(C'', \tau'')$
- If $v = 1$, return $M$
- else return $\perp$. 
## Encrypt-then-MAC Security

<table>
<thead>
<tr>
<th>Security</th>
<th>Weak MAC</th>
<th>Strong MAC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Privacy</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IND-CPA</td>
<td>Secure</td>
<td>Secure</td>
</tr>
<tr>
<td>IND-CCA</td>
<td>Insecure</td>
<td>Secure</td>
</tr>
<tr>
<td>NM-CPA</td>
<td>Insecure</td>
<td>Secure</td>
</tr>
<tr>
<td>Integrity</td>
<td></td>
<td></td>
</tr>
<tr>
<td>INT-PTXT</td>
<td>Secure</td>
<td>Secure</td>
</tr>
<tr>
<td>INT-CTXT</td>
<td>Insecure</td>
<td>Secure</td>
</tr>
</tbody>
</table>
# Summary of Methods

## Weakly Unforgeable

<table>
<thead>
<tr>
<th>Composition Method</th>
<th>Privacy</th>
<th>Integrity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Encrypt-and-MAC</td>
<td>Insecure</td>
<td>Secure</td>
</tr>
<tr>
<td>MAC-then-Encrypt</td>
<td>Secure</td>
<td>Secure</td>
</tr>
<tr>
<td>Encrypt-then-MAC</td>
<td>Secure</td>
<td>Secure</td>
</tr>
</tbody>
</table>

## Strongly Unforgeable

<table>
<thead>
<tr>
<th>Composition Method</th>
<th>Privacy</th>
<th>Integrity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Encrypt-and-MAC</td>
<td>Insecure</td>
<td>Secure</td>
</tr>
<tr>
<td>MAC-then-Encrypt</td>
<td>Secure</td>
<td>Secure</td>
</tr>
<tr>
<td>Encrypt-then-MAC</td>
<td>Secure</td>
<td>Secure</td>
</tr>
</tbody>
</table>
Theorem 4.7

- Encrypt-then-MAC method is IND-CPA and INT-PTXT
- SE be a symmetric scheme
- MA be message authentication scheme

\[
\text{Adv}_{SE}^{\text{ind-\text{cpa}}} (k, t, q, \mu) \leq \text{Adv}_{SE}^{\text{ind-\text{cpa}}} (k, t, q, \mu)
\]

\[
\text{Adv}_{SE}^{\text{int-\text{ptxt}}} (k, t, q_e, q_d, \mu_e, \mu_d) \leq \text{Adv}_{MA}^{\text{wuf-\text{cma}}} (k, t, q_e, q_d, \mu_e, \mu_d)
\]
Theorem 4.7 - IND-CPA

\[ \text{Adv}^{\text{ind-cpa}} \left( k \right) \leq \text{Adv}^{\text{ind-cpa}} \left( k, t, q, \mu \right) \]

**Adversary** \( A_{k_e(\text{LR}(\ldots,b))}^k (\kappa) \)

\[ k_m \leftarrow R \quad K_m (\kappa) \]

For \( i = 1, \ldots, q \) do

When \( A \) makes a query \( (M_{i,0}, M_{i,1}) \) to its left – or – right encryption oracle do

\[ C_i \leftarrow E_{k_e(\text{LR}(M_{i,0}, M_{i,1}, b))}; \tau_i \leftarrow T_{K_m} (C_i); A \leftarrow C_i \parallel \tau_i \]

\( A \Rightarrow b' \)

Return \( b' \)
Theorem 4.7 - INT-PTXT

\[ \text{Adv}^{\text{int-}ptxt}_{SE,A} (k) \leq \text{Adv}^{wuf-cma}_{M,A_P} (k) \]

Adversary \( F_p^{T_{K_m} (\cdot), V_{K_m} (\cdot)} (\kappa) \)

\( k_e \leftarrow R_{K_e (\kappa)} \)

For \( i = 1, \ldots, q_e + q_d \) do

When A makes a query \( M_i \) to its encryption oracle do

\( C_i \leftarrow E_{K_e} (M_i); \tau_i \leftarrow T_{K_m} (C'_i); A \leftarrow C_i \parallel \tau_i \)

When A makes a query \( C_i \) to its verification oracle do

Parse \( C_i \) as \( C_i \parallel \tau_i; v_i \leftarrow V_{K_m} (C_i, \tau_i); A \leftarrow v_i \)
Proposition 4.9

- Encrypt-then-MAC method with a SUF-CMA-secure MAC is INT-CTXT, IND-CPA, and IND-CCA

\[ \text{Adv}_{SE,A}^{\text{int-ctxt}}(k) \leq \text{Adv}_{MA,F}^{\text{suf-cma}}(k) \]

\[ \text{Adv}_{SE}^{\text{ind-cpa}}(k, t, q, \mu) \leq \text{Adv}_{SE}^{\text{ind-cpa}}(k, t, q, \mu) \]

\[ \text{Adv}_{SE}^{\text{int-ctxt}}(k, t, q_e, q_d, \mu_e, \mu_d) \leq \text{Adv}_{MA}^{\text{suf-cma}}(k, t, q_e, q_d, \mu_e + q_e l, \mu_d) \]

\[ \text{Adv}_{SE}^{\text{ind-cca}}(k, t, q_e, q_d, \mu_e, \mu_d) \leq 2 \times \text{Adv}_{MA}^{\text{suf-cma}}(k, t, q_e, q_d, \mu_e + q_e l, \mu_d) + \text{Adv}_{SE}^{\text{ind-cpa}}(k, t, q_e, \mu_e) \]
Conclusion

Encrypt-then-MAC provides the most secure solution for authenticated encryption.
CBC – Cipher Block Chain

- If IV is different then instances of same msg (or block) will be encrypted differently
- If K’th cipher block $C_k$ gets corrupted in transmission – only blocks $P_k$ and $P_{k+1}$ are affected
  - This can also allow some msg tampering
- If one plaintext block $P_k$ is changed – All subsequent ciphertext blocks will be affected
  - This leads to an effective MAC
If the same key is used then identical plaintext blocks map to identical ciphertext.
Proposition 4.1

- Encrypt-and MAC method is not IND-CPA
Proposition 4.2

- Encrypt-and MAC method is IND-CPA insecure for any deterministic MAC)
Theorem 4.3

- Encrypt-and-MAC is INT-PTXT secure
Proposition 4.4

- Encrypt-and-MAC method is not INT-CTXT secure
Theorem 4.5

- MAC-then-encrypt method is both INT-PTXT and IND-CPA secure
Proposition 4.6

- MAC-then-encrypt method is not NM-CPA secure
Proposition 4.8

- Encrypt-then-MAC method with a WUF-CMA-secure MAC is not NM-CPA secure