COS433/Math 473: Cryptography

Mark Zhandry
Princeton University
Fall 2020
Announcements/Reminders

PR1 Due TODAY

HW3 due on Oct 20
Previously on COS 433...
Message Integrity
Recall: CPA Security

\[ m_0, m_1 \in M_\lambda \]

Challenger

\[ k \leftarrow K_\lambda \]

\[ c \leftarrow \text{Enc}(k, m_b) \]

\[ \text{LoR-Exp}_b(\text{?, } \lambda) \]
Limitations of CPA security

How?
Message Authentication

Goal: If Eve changed $m$, Bob should reject
Message Authentication Codes

Syntax:
- Key space $K_\lambda$
- Message space $M_\lambda$
- Tag space $T_\lambda$
- $MAC(k,m) \rightarrow \sigma$
- $Ver(k,m,\sigma) \rightarrow 0/1$

Correctness:
- $\forall m,k, \ Ver(k,m,\ MAC(k,m) ) = 1$
1-time Security For MACs

\[
\text{Output 1 iff:} \\
\begin{align*}
\text{• } m^* &\neq m \\
\text{• } \text{Ver}(k, m^*, \sigma^*) &\neq 1
\end{align*}
\]

\[1\text{CMA-Adv}(\cdot, \lambda) = \Pr[\text{outputs 1}]\]
Definition: \((MAC,\text{Ver})\) is 1-time statistically secure under a chosen message attack (statistically \(1\text{CMA-secure}\)) if, for all \(\lambda\), \(\exists\) negligible \(\varepsilon\) such that:

\[
1\text{CMA-Adv}(\text{Ver}, \lambda) \leq \varepsilon(\lambda)
\]
Today

Message Integrity, continued
Authenticated encryption
Question

Is perfect security ($\varepsilon=0$) possible?
A Simple 1-time MAC

Suppose $H_\lambda$ is a family of pairwise independent functions from $M_\lambda$ to $T_\lambda$

For any $m_0 \neq m_1 \in M_\lambda$, $\sigma_0, \sigma_1 \in T_\lambda$

$\Pr_{h \leftarrow H_\lambda} [ h(m_0) = \sigma_0 \land h(m_1) = \sigma_1 ] = 1/|T_\lambda|^2$

$K = H_\lambda$

$MAC(h, m) = h(m)$

$Ver(h, m, \sigma) = (h(m) == \sigma)$
Theorem: If $|T_{\lambda}|$ is super-polynomial, then $(MAC, Ver)$ is 1-time secure

Intuition: after seeing one message/tag pair, adversary learns nothing about tag on any other message

So to have security, just need $|T_{\lambda}|$ to be large
Ex: $T_{\lambda} = \{0,1\}^{128}$
Constructing Pairwise Independent Functions

\[ T_\lambda = \mathbb{F} \text{ (finite field of size } \approx 2^\lambda) \]
- Example: \( \mathbb{Z}_p \) for some prime \( p \)

Easy case: let \( M_\lambda = \mathbb{F} \)
- \( H_\lambda = \{ h(x) = a \times + b: a, b \in \mathbb{F} \} \)

Slightly harder case: Embed \( M_\lambda \subseteq \mathbb{F}^n \)
- \( H_\lambda = \{ h(x) = \langle a, x \rangle + b: a \in \mathbb{F}^n, b \in \mathbb{F} \} \)
Multiple Use MACs?

Just like with OTP, if use 1-time MAC twice, security no longer guaranteed

Why?
q-Time MACs

Output 1 iff:

- \( m^* \notin \{m_1, \ldots, m_q \} \)
- \( \text{Ver}(k, m^*, \sigma^*) = 1 \)

\[
q\text{CMA-Adv}(\mathcal{A}, \lambda) = \Pr[\text{outputs 1}] = 1
\]
Definition: \( (\text{MAC,Ver}) \) is \( q \)-time statistically secure under a chosen message attack (statistically \( q \text{CMA-secure} \)) if, for all \( \exists \) making at most \( q \) queries, \( \exists \) negligible \( \varepsilon \) such that:

\[
\text{CMA-Adv}(\bullet, \lambda) \leq \varepsilon(\lambda)
\]
Constructing $q$-time MACs

Ideas?

Limitations?
Impossibility of Large $q$

Theorem: Any $\textsf{qCMA}$-secure MAC must have $q \leq \log |K_\lambda|$
Proof

Idea:
• By making $q \gg \log |K_\lambda|$ queries, you should be able to uniquely determine key
• Once key is determined, can forge any message

Problem:
• What if certain bits of the key are ignored
• Intuition: ignoring bits of key shouldn’t help
Proof

Define $r_q$ as follows:

- Challenger chooses random key $k$
- Adversary repeatedly choose random (distinct) messages $m_i$ in $M_\lambda$
- Query the CMA challenger on each $m_i$, obtaining $\sigma_i$
- Let $K'_q$ be set of keys $k'$ such that $\text{MAC}(k',m_i)=\sigma_i$ for $i=1,...,q$
- Let $r_q$ be the expected size of $K'_q$
Claim: If \((\text{MAC,Ver})\) is qCMA-secure, then 
\[ r_q \leq r_{q-1}/2 \]

If not, then with probability at least \(1/4\),
\[ |K'_q| > |K'_{q-1}|/4 \]

Attack:
- Make \(q-1\) queries on random messages \(m_i\)
- Choose key \(k\) from \(K'_{q-1}\)
- Choose random \(m_q\), compute \(\sigma_q = \text{MAC}(k,m_q)\)
- Output \((m_q, \sigma_q)\)

Probability of forgery?
Claim: If \((\text{MAC,Ver})\) is qCMA-secure, then 
\[ r_q \leq r_{q-1}/2 \]

Finishing the impossibility proof:

• \( r_q \) is always at least 1 (since there is a consistent key)

• \( r_0 = |K_\lambda| \)

• \( 1 \leq r_q \leq r_0/2^q \leq |K_\lambda|/2^q \)

• Setting \( q > \log |K_\lambda| \) gives a contradiction
Computational Security

Definition: \((\text{MAC},\text{Ver})\) is computationally secure under a chosen message attack (\text{CMA-secure}) if, for all \(\mathcal{A}\) running in polynomial time (and making a polynomial number of queries), \(\exists\) negligible \(\varepsilon\) such that

\[
\text{CMA-Adv(\mathcal{A}, \lambda)} \leq \varepsilon(\lambda)
\]
Constructing MACs

Use a PRF

\[ F : K_\lambda \times M_\lambda \rightarrow T_\lambda \]

\[ MAC(k,m) = F(k,m) \]

\[ Ver(k,m,\sigma) = (F(k,m) == \sigma) \]
Theorem: If $F$ is a secure PRF and $|T_\lambda|$ is super-polynomial, then $(\text{MAC,Ver})$ is CMA secure.
Security Proof

Assume toward contradiction polynomial time

Hybrids!
Security Proof

Hybrid 0

\[ m_i \in M_\lambda \]

\[ \sigma_i \]

\[ (m^*, \sigma^*) \]

\[ k \leftarrow K_\lambda \]

\[ \sigma \leftarrow F(k, m_i) \]

Output 1 iff:
- \[ m^* \notin \{m_1, \ldots\} \]
- \[ F(k, m^*) = \sigma^* \]

CMA Experiment
Security Proof

Hybrid 1

\[ m_i \in M_\lambda \]

\[ \sigma_i \]

\[ (m^*, \sigma^*) \]

\[ H \leftarrow \text{Funcs}(M_\lambda, T_\lambda) \]

\[ \sigma \leftarrow H(m_i) \]

Output 1 iff:

- \( m^* \notin \{m_1, \ldots\} \)
- \( H(m^*) = \sigma^* \)
Security Proof

Claim: in Hybrid 1, output 1 with probability $1/|T_\lambda|$

- sees values of $H$ on points $m_i$
- Value on $m^*$ independent of ‘s view
- Therefore, probability $\sigma^* = H(m^*) = 1/|T_\lambda|$
Security Proof

Claim: \( |\Pr[1 \leftarrow \text{Hyb1}] - \Pr[1 \leftarrow \text{Hyb2}]| \leq \varepsilon(\lambda) \)

Suppose not, construct PRF adversary
MACs/PRFs for Larger Domains

We saw that block ciphers are good PRFs

However, the input length is generally fixed
• For example, AES maximum block length is 128 bits

How do we handle larger messages?
Block-wise Authentication?

Why is this insecure?
Block-wise Authentication?

Why is this insecure?
Block-wise Authentication?

Why is this insecure?
Block-wise Authentication?

$r$ a random nonce

$r\begin{array}{c}0 \leftrightarrow k & 1 \leftrightarrow k & 2 \leftrightarrow k & 3 \leftrightarrow k & 4 \leftrightarrow k & 5 \leftrightarrow k\end{array}\) $)

Secure, but not very useful in practice
Theorem: CBC-MAC is a secure PRF for \textit{fixed-length} messages
Timing Attacks on MACs

How do you implement check $F(k,m) == \sigma$?

String comparison often optimized for performance

**Compare(A,B):**
- For $i = 1,\ldots,A.length$
  - If $A[i] != B[i]$, abort and return False;
- Return True;

Time depends on number of initial bytes that match
Timing Attacks on MACs

To forge a message \( \mathbf{m} \):

For each candidate first byte \( \sigma_0 \):
- Query server on \((\mathbf{m}, \sigma)\) where first byte of \( \sigma \) is \( \sigma_0 \)
- See how long it takes to reject

First byte is \( \sigma_0 \) that causes the longest response
- If wrong, server rejects when comparing first byte
- If right, server rejects when comparing second
Timing Attacks on MACs

To forge a message $m$:

Now we have first byte $\sigma_0$

For each candidate second byte $\sigma_1$:
• Query server on $(m, \sigma)$ where first two bytes of $\sigma$ are $\sigma_0, \sigma_1$
• See how long it takes to reject

Second byte is $\sigma_1$ that causes the longest response
Holiwudd Criptoe!

Most likely not what was meant by Hollywood, but conceivable
Thwarting Timing Attacks

Possibility:
• Use a string comparison that is guaranteed to take constant time
• Unfortunately, this is hard in practice, as optimized compilers could still try to shortcut the comparison

Possibility:
• Choose random block cipher key $k'$
• Compare by testing $F(k', A) == F(k', B)$
• Timing of “==” independent of how many bytes $A$ and $B$ share
Alternate security notions
Strongly Secure MACs

\[
m_i \in M \quad \Rightarrow \quad \sigma_i = \text{MAC}(k, m_i)
\]

Output 1 iff:
- \((m^*, \sigma^*) \notin \{(m_1, \sigma_1), \ldots\}
- \text{Ver}(k, m^*, \sigma^*) = 1

\[
\text{SCMA-Adv}(\text{Bob}) = \Pr[\text{Bob outputs 1}]
\]
Strongly Secure MACs

Useful when you don’t want to allow the adversary to change *any* part of the communication

If there is only a single valid tag for each message (such as in the PRF-based MAC), then (weak) security also implies strong security

In general, though, strong security is stronger than weak security
Adding Verification Queries

\[ k \leftarrow K \]
\[ \sigma_i \leftarrow \text{MAC}(k, m_i) \]
\[ b \leftarrow \text{Ver}(k, m, \sigma) \]

Output 1 iff:
- \( m^* \notin \{m_1, \ldots \} \)
- \( \text{Ver}(k, m^*, \sigma^*) = 1 \)

\[ \text{CMA}'-\text{Adv}(\cdot) = \Pr[\ \text{ outputs } 1] \]
Theorem: \((\text{MAC,Ver})\) is strongly CMA secure if and only if it is strongly CMA’ secure
Improving efficiency
Limitations of CBC-MAC

Many block cipher evaluations

Sequential
Carter Wegman MAC

\[ k' = (k,h) \]

where:

- \( k \) is a PRF key for \( F:K \times R \rightarrow Y \)
- \( h \) is sampled from a pairwise independent function family

**MAC(k',m):**

- Choose a random \( r \leftarrow R \)
- Set \( \sigma = (r, F(k,r) \oplus h(m)) \)
Theorem: If \( F \) is secure and \( |T|, |R| \) are super-polynomial, then the Carter Wegman MAC is strongly CMA secure.
Efficiency of CW MAC

\textbf{MAC}(k',m):
- Choose a random $r \leftarrow \mathcal{R}$
- Set $\sigma = (r, F(k,r) \oplus h(m))$

\textbf{h} much more efficient than PRFs

PRF applied only to small nonce $r$
\textbf{h} applied to large message $m$
PMAC: A Parallel MAC

\[
\begin{align*}
&k' \oplus k' \oplus k' \oplus k' \oplus k' \\
&F \quad F \quad F \quad F \quad F \\
&k \quad k \quad k \quad k \quad k \\
&\sigma
\end{align*}
\]
Announcements/Reminders

PR1 Due TODAY

HW3 due on Oct 20