Lecture 6: Encryption in Practice
Plan

- Recap: AES-CTR
- Three constructions
  - AES
  - DES
  [Stretch break]
  - Chacha20
- MITM attack on 2DES

Logistics

* Peer 1 due Friday
* Friday recitation is all about project. Important!
* Next week: meet TA/instructor 12: project details OH
* Think on groups for project.
Encryption used everywhere?

- Phonex
- Computer
- Satellite

\{ Essentially any net com today \}

- NIST publishes standards for encryption
  - Widely used, read often for selling to govt
  - Not only standards org (IETF also, ISO, ...)

- NIST Ciphers
  - DES (1975) \[ 128 \times 2^{36} \]
  - 3DES \[ 128 \times 2^{168} \]
  - AES (1998) \[ 128 \times \{ 2^{128}, 2^{192}, 2^{256} \} \]

\{ Still used! \}

N.B. DES key size is far too small.

\{ +SECRET: AES-128/192/256 \}
\{ TOP SECRET: AES-192/256 \}

\{ A2s are public! \}

Are PRPs, but some common primitives (e.g. ChaCha20) are not

- While you will never need to implement these primitives yourself, worthwhile to understand design.

- Hash functions are coming up, but not today...
Recap: AES-GCM (Authenticated encryption)

CPA-secure encryption
Secure MAC

\[ \implies \text{Authenticated encryption} \Rightarrow \text{CCA security} \]

Encrypt then MAC.

Uses AES as PRF: \( F: \mathbb{F}_{2^n} \to \{0,1\}^n \) \( (n=128) \)

\( m = m_1 \oplus m_2 \oplus \ldots \oplus m_n \)

\[ \begin{align*}
C_0 &= \text{Ciphertext} \\
\text{tag} &= F(k, IV) \oplus \sum_{i=1}^{n} C_i \cdot r^{n-i}
\end{align*} \]

(Also need to include msg length in hash but I'm omitting it for simplicity.)

\( r = F(k, "0000-0") \)

\( \implies \) Single pass over the message

\( \implies \) Careful use of PRF lets use same key for encryption & MAC

Not so safe integrally
Other notes on AES-GCM

- CPUs have HW support for AES (GBs per second) (AES-NI)
  \[ \Rightarrow \text{Essentially "for free" today.} \]

- As we discussed, AES is PRP but used here as a PRE.
  Why is that safe?! PRP≠PRE

**"PRF Switching Lemma" (See Borel-Shoup)**

Let \( P: \mathbb{G} \times \{0,1\}^* \rightarrow \{0,1\}^n \) be a PRF.

Then for any PRP adv \( \mathcal{A}_{\text{PRP}} \), \( \exists \) PRF adv \( \mathcal{A}_{\text{PRF}} \) s.t.

\[ |\mathcal{A}_{\text{PRP}} - \mathcal{A}_{\text{PRF}}| \leq \frac{q^2}{2^{n+1}}. \]

**Intuition:**

* Collisions in outputs is only diff by PRF & PRP
* Until \( q < 2^{\frac{n}{2}} \) will not expect to see collisions by Birthday paradox
* After that, can distinguish

\[ \Rightarrow \text{"Sweet 32 attack" 7GB GB traffic on 3DES (64-bit block)} \]
Properties that AES-GCM doesn't provide

- **Nonce-reuse protection**
  - Some modes of operation do (at some cost)
    - Reusing nonce reveals equality & nothing more

- **Commitment**
  - Can find \((k_1, k_2, c)\) s.t.
    - \(\text{Dec}(k_1, c) = \text{msg}_1\)
    - \(\text{Dec}(k_2, c) = \text{msg}_2\)

\[
\begin{align*}
  k_1 &\rightarrow m_1 \\
  k_2 &\rightarrow m_2
\end{align*}
\]

\(\text{Control} \approx \frac{1}{2} \text{bits of each msg}\)

[ Dodis, Cribbs, Ristenpart, Woodage '18 ]
**Why MAC-then-encrypt is bad:**

* Some enc schemes (CBC mode) require plaintext be multiple of block size, e.g. 16 bytes
  - Convenient & sometimes necessary
* Pad msg with n indicating "truncate n+1 bytes"

\[ \text{msg} \rightarrow \text{MAC} \rightarrow \text{pad} \]

\[ \text{msg} \mid \text{tag} \mid 3 \ 3 \ 3 \ 3 \]

\[ \text{pseudorandom bytes from AES}(k) \]

\[ \text{ct} = \]

\[ 3 \ 3 \ 3 \ 3 \]

\[ 4 \ 4 \ 4 \ 4 \ 4 \]

\[ \text{evil } \text{ct} \]

\[ \text{msg} \mid \text{tag} \mid 4 \ 4 \ 4 \ 4 \ 4 \]

If adv can learn whether padding is valid, learns one byte of msg! \( \rightarrow \) Timing, error msg etc.
Three constructions

AES - Substitution permutation
DES - Feistel network
Chacha20 - "Even-Mansour" (?) PRF

Some of crypto is based on "nice" assumptions
→ "win win". E.g., Factoring
→ Nice things cost too much

PRF/PRP design is messier in some ways:
* Design to resist best known attacks
* Try to get others to break (NIST competitions)
* Patch when broken

→ Surprise: No serious break of 3DES (beyond obvious ones)

Difficult part (in some sense) isn't security, it's getting security with good performance on all hw.
(example of 3rd gradu)
Warning!

Do not attempt to build or implement a block cipher (or mode of operation yourself!) by timing & cache attacks, cryptanalysis, etc. It takes many years of effort to gain confidence in design.
Design of AES (PRP/block cipher)

AES is an "Iterated Even-Mansour cipher"

Uses invertible \( \Pi : \{0,1\}^{128} \rightarrow \{0,1\}^{128} \)

* Very simple - substitution, linear ops, etc
  (Subbytes, ShiftRows, MixColumns)

Derived from key using invertible linear fn

\[ \begin{align*}
  k_0 & \rightarrow x + \Pi \\
  k_1 & \rightarrow \Pi + \cdots \\
  k_9 & \rightarrow \Pi^* + y
\end{align*} \]

AES 128 has "10 rounds" 14 rounds

Security justification

* After two decades of cryptanalysis, no great attacks
* If we model \( \Pi \) as a random perm \( \Rightarrow \) Prove security.
Stretch
Break
DES cipher (PRP/block cipher)

* Horst Feistel (MIT undergrad) \to \text{go} \to \text{IBM}
* Lucifer — precursor to DES, was Feistel net
* Also how you get PRF \Rightarrow \text{PRP}

\[
x_0 \quad \text{y}_0
\]

\[
\rightarrow \quad f_i(k,.) \quad \rightarrow
\]

\[
x_1 \quad \text{y}_1
\]

\[
\rightarrow \quad f_2(k,.) \quad \rightarrow
\]

\[
x_2 \quad \text{y}_2
\]

---

Invertible?

\[
y_{i-1} = x_i \quad x_{i-1} = S(y_i \oplus S(x_i))
\]

* Luby & Rackoff showed that if \( S(k,:) \) is a secure PRF
  \( \Rightarrow \text{Feistel}^2 \) is a secure PRP
  \( \text{[Not obvious]} \)

* In practice (e.g. DES) \( S \) is NOT a PRF
  \( \Rightarrow \) But LR analysis gives some justification for design

* In DES shared many features w/ AES
  round function (Substitution, permutation)
ChaCha20 (PRF)

- Essentially a PRF (used as "stream cipher")
  
  key = 256 bits F: \{0,1\}^{256} \times \{0,1\}^{128} \rightarrow \{0,1\}^{512}

- Design rationale
  * Used in CTR mode for CPA-secure encryption.
  * The permutation \( \Pi \) performs 10 rounds of simple bit operations on a 4x4 matrix of 32-bit words (add, rot, xor).
3DES

- DES 56-bit key too short.
- EFF DES Cracker: 1998: $25k of compute
  - Now $20, takes a few days

\[ 3DES((k_1, k_2, k_3), m) = DES(k_3, DES^{-1}(k_2, DES(k_1, m))) \]

(Clever hack: \( k_1 = k_2 = k_3 \Rightarrow 3DES = DES \))

* Keylen is 168 bits
* MITM takes \( \approx 2^{113} \) time.
Broken idea: 2DES
- "Meet-in-the-middle" attack. 
  La shows up all over the place.

\[
2DES((k_1, k_2), m) = DES(k_2, DES(k_1, m)).
\]
- Key is \(56 \times 2 = 112\) bits
- Problem: Meet-in-the-middle attack
  Say attacker gets \((m_0, c_o)\) st. \(c_i = DES(k^*_2, DES(k^*_1, m_i))\)

By birthday paradox, expect to find a collision after \(\sqrt{2^{112}} = 2^{56}\) time. Space = \(2^{64}\) Can reduce?

\[\Rightarrow\text{Keylen is only effectively 56 bits... no improvement.}\]
How does one break a PRF/PRP?

Linear cryptanalysis.

\[ \Pr_{p,c} \left[ P_1 \oplus P_3 \oplus P_5 \oplus C_1 \oplus C_2 \oplus C_7 = k_1 \oplus k_5 \oplus k_{12} \right] > \frac{1}{2} + \varepsilon \]

Matsui (1993) found linear relation like this with \( \varepsilon = 2^{-21} \)

Attack:
- Find \( \frac{1}{\varepsilon^2} = 2^{42} \) (pt, ct) pairs
- Compute noisy guesses of key bits using (*)
- After \( \frac{1}{\varepsilon^2} \) pt/ct's, will get correct key bits whp.
  \( \Rightarrow \) Reveals \( \approx 15 \) key bits. Brute force the rest.

\( \Rightarrow \) Small bias causes serious break \( 2^{56} \rightarrow 2^{42} \)