April 12: Symmetric-Key Primitives

Review from Last Time

Definitions/Spinach

* Negligible functions, e.g., alg's, see parameters
* PRG $G : \{0,1\}^n \rightarrow \{0,1\}^{g(n)}$ (for $\ell(n) > n$)

\[ \{ s \in \{0,1\}^n \mid G(s) \} \subseteq \{ z \in \{0,1\}^{g(n)} \} \]

"Stretch a short random seed into long random-looking string",

\[ \text{Blum Micali: PRG (one-bit PRG \Rightarrow poly(n)-bit PRG)} \]

\[ \text{Hybrid argument} \]

* PRF $F : \mathcal{X} \times \mathcal{X} \rightarrow \mathcal{Y}$

\[ \{ k \in \mathcal{X} : F(k, \cdot) \} \subseteq \{ F \in \mathcal{F} \mid \text{Exp}[\mathcal{X}, \mathcal{Y}] : \mathcal{S}(\cdot) \} \]

"Indist. from a random function"

* PRP $F : \mathcal{X} \times \mathcal{X} \rightarrow \mathcal{X}$

\[ \{ k \in \mathcal{X} : F(k, \cdot) \} \subseteq \{ F \in \text{Perey} \mid \text{Exp}[\mathcal{X}] : \mathcal{S}(\cdot) \} \]

\[ \text{GGM tree} \quad \text{PRG} \quad \text{PRP} \quad \text{PRF} \quad \text{Counter mode} \quad \text{Luby-Rackoff 3-round Feistel} \]

\[ \text{If you have any, you have all (up to polynomial efficiency/security loss)} \]

Questions?

One application of PRPs:

"Format-preserving encryption:

encrypt \{ credit card \# \} \rightarrow \{ credit card \# \}
Logistics - April 12

- HW 0 due now
- Project proposal due next week
  * ideas online
  * talk to us at OH/after (ies)
- Lecture scrib? (1st/2nd half)
- Note cards
Cycle Walking

I claimed that PRPs are useful for encrypting
\( \{ \text{cred + card #s} \} \rightarrow \{ \text{cred + card #s} \} \)
but Feistel network only gives a PRP on n-bit strings.

Problem: The set \( C \) of valid cc #5s is a smallish subset of \( \{0, 1\}^n \). Not all 16-digit #s are cc #5s!
For example, cc #s have a "check digit" / CRC. We want a PRP on cc #5s!
Say we have a predicate
\[
\text{Valid: } \{0, 1\}^n \rightarrow \{0, 1\}
\]
Define \( C = \{ x | \text{Valid}(x) = 1 \} \subseteq \{0, 1\}^n \)

We have PRP \( F : 2^n \times \{0, 1\}^n \rightarrow (0, 1)^n \) (PRP on strings)
We want PRP \( F_c : 2^n \times C \rightarrow C \). (PRP on cc #5s)

Claim (Shnappel, Orman):
Given \( F, \text{Valid}, \) can construct \( F_c \) as long as \( \frac{1}{2^n} > \text{poly}(n) \).

Pf Idea: By picture

\[\{0, 1\}^n \]
\[F_c(k, x)\]
\[y \leftarrow \text{Valid}(x)\]
\[\text{do } y \leftarrow F(y) \]
\[\text{while Valid}(y) \neq 1\]
\[\text{return } y\]

\(F^+(\cdot, \cdot)\) is similar.
Cycle Walking

Correctness The expected # of invocations of $F$ is $\frac{2^n}{121} \leq \text{poly}(n)$.

As long as "valid" $x$s are not too sparse, this is ok.

Security As with Luby-Rackoff analysis, it's a two step process...

1) Replace $F$ with a random permutation $\Pi \in \text{Perm}[2^n]$.
2) Argue that cycle-walking using $\Pi$ gives a random permutation $\Pi_c : C \rightarrow C$

Possible problem: side-channel attacks!
Time to encrypt depends on $msg$.
Padding?
Or: you have better idea?
Even-Mansour Cipher

As David mentioned, crypto (PRF, key, etc.) requires P≠NP and more, so we can't unconditionally construct PRPs/PRFs in standard computational models.

What do we do?
1) Make assumptions (e.g., Factoring is hard)
2) Change the model

Last week, we showed how to construct P2P under assumption that we have a one-bit PRF. (Approach #1)

Now we will show how to use approach #2 to construct a PRP.

Random Permutation Model

"Standard Model"
Both good guy and adv.
use a Turing machine

"Random Permutation Model" (RPM)

Today
⇒ In RPM, we can construct unconditionally secure PRPs!

The catch:
1) \( \mathbf{T} \) is exponentially large!
2) Schemes secure in RPM may be broken when \( \mathbf{T} \) is instantiated with a real efficiently computable perm.

Skill! Useful analysis technique: IS broken in RPM, definitely broken in standard model!
* Shows that any attack will have to exploit structure of permutation
N.B. Random oracle model is like RPM with random function & ideal cipher model is random family of f,

Even Manger

\[ \Pi \text{ is R.P., } \Pi : \{0,1\}^n \rightarrow \{0,1\}^n. \]
\[ E : \{0,1\}^n \times \{0,1\}^n \rightarrow \{0,1\}^n \]
\[ E(k, m) = k \oplus \Pi^{-1}(k \oplus m) \text{ (two diff keys) } \]
\[ D(k, c) = \Pi(k \oplus \Pi^{-1}(k \oplus c)) \]
\[ = k \oplus \Pi^{-1}(k \oplus \Pi(k \oplus c)) \]
\[ = k \oplus \Pi^{-1}(k \oplus m) \]
\[ = k \oplus k \oplus m \]
\[ = m \]

Interesting because...

- It's as simple as it gets.
- Basis for design of AES (SPN)
- AES uses 10-rounds of SPN
- Gives a "rigorous heuristic" for design of PAPs in practice
Even - Mansour

- Paper probes two non-standard sec properties
- Boreh-Shoup (Monts, Kilian - Koga-ray '00) show $E_m$ is a PRP.

Thm (Kilian, Koga-ray) (4.1 in Boreh - Shoup)

Let $A_1^{\Pi, \Pi}$ be an adversary making at most $Q_{\Pi}$ queries and at most $Q_{\Pi}$ permutation queries, then

$$PRPAdv^{\Pi}_{E_m, A} \leq \frac{2 \cdot Q_{\Pi} \cdot Q_{\Pi}}{2^n}.$$  

N.B. We prove security against advs running in unbounded time.

\rightarrow Info-theoretic result.
Proof

Overview

Game 0: Challenger responds using EM cipher.
Game 1: Rephrasing of game 0.
Game 2: Challenger responds with independent permutation $T_E(\cdot)$.

Define $p_i = \Pr[\text{Adv output } i \text{ in game } i]$.

Want to show $|p_0 - p_1| \leq 2^{Q_{\text{enc}} + Q_{\text{m}}} / 2^n$.

In excruciating detail... the proof (good to see one).

The challenger responds to $E, T_1, T_2^*$ queries.

Rather than fixing random $T_1$ in advance, chal builds it up "lazily" in response to queries.

We represent $T_2$ as a set of $(\omega, p)$ pairs.
Game 0: (Real Construction)

Setup: \( \Pi \leftarrow \emptyset \)
\[ k \leftarrow \mathcal{R} \{0,1\}^n \]

\( E \) query on \( m \):
\[ \alpha \leftarrow m \oplus k \]
if \( \Pi(\alpha) \) undefined
\[ \Pi(\alpha) \leftarrow \{0,1\}^n \times \text{Range}(\Pi) \]
return \( k \oplus \Pi(\alpha) \)

\( \overline{\Pi} \) query on \( \alpha \):
if \( \Pi(\alpha) \) undefined
\[ \Pi(\alpha) \leftarrow \{0,1\}^n \times \text{Range}(\Pi) \]
return \( \Pi(\alpha) \)

\( \Pi^{-1} \) query on \( \beta \):
if \( \Pi^{-1}(\beta) \) undefined
\[ \Pi^{-1}(\beta) \leftarrow \{0,1\}^n \setminus \text{Domain}(\Pi) \]
return \( \alpha \)

Game 1 (Intermediate)

Setup: \( \Pi_E \leftarrow \emptyset \) \( \leftarrow \) Private random Perm in ARP game
\( \Pi \leftarrow \emptyset \) \( \leftarrow \) Public random Perm
\[ k \leftarrow \mathcal{R} \{0,1\}^n \]

\( E \) query on \( m \):
\[ \alpha \leftarrow m \oplus k \]
if \( \Pi_E(\alpha) \) and \( \Pi(\alpha) \) undefined
\[ \Pi_E(\alpha) \leftarrow \{0,1\}^n \times \text{Range}(\Pi_E) \]
\[ \Pi(\alpha) \leftarrow \{0,1\}^n \times (\text{Range}(\Pi) \cup \text{Range}(\Pi_E)) \]
return \( k \oplus \Pi(E(\alpha)) \)

\( \Pi \) query on \( \alpha \):
if \( \Pi(\alpha) \) and \( \Pi_E(\alpha) \) undefined
\[ \Pi(\alpha) \leftarrow \{0,1\}^n \setminus \text{Range}(\Pi) \]
\[ \Pi(\alpha) \leftarrow \{0,1\}^n \setminus (\text{Domain}(\Pi) \cup \text{Domain}(\Pi_E)) \]
return \( \Pi(\alpha) \)

\( \Pi^{-1} \) query on \( \beta \):
if \( \Pi^{-1}(\beta) \) and \( \Pi_E^{-1}(\beta) \) undefined
\[ \Pi^{-1}(\beta) \leftarrow \{0,1\}^n \setminus \text{Domain}(\Pi) \]
\[ \Pi^{-1}(\beta) \leftarrow \{0,1\}^n \setminus (\text{Domain}(\Pi) \cup \text{Domain}(\Pi_E)) \]
return \( \Pi^{-1}(\beta) \)
Proof (cont'd)

In Game 0: $E / \Pi / \Pi^*$ queries give responses as in real construction.

In Gam 1: Everything is exactly as in Game 0, except that we split $\Pi$ into two parts: $\Pi$ and $\Pi_E$.

$$\rightarrow \text{By construction: } \Pi \cap \Pi_E = \emptyset, \text{ so } \Pi \text{ and } \Pi_E \text{ together define a permutation.}$$

In Game 2: $E$ queries responded to with $\Pi_E$

$\Pi/\Pi^*$ queries responded to with independent $\Pi$

$$\Rightarrow E \text{ is a real random permutation.}$$

$|p_0 - p_1| = 0$ since games are identical.

To complete proof, want to bound $|p_1 - p_2|$.

Bad event $B$ is in Game 2 that

$$\text{Domain}(\Pi) \cap \text{Domain}(\Pi_E) \neq \emptyset \text{ or } \text{Range}(\Pi) \cap \text{Range}(\Pi_E) \neq \emptyset.$$

As long as $B$, Games 1 and 2 are identical. (See Boolean-Sharp for details.)

Intuitively: Bad event is that $\Pi$ and $\Pi_E$ conflict—they give contradictory answers at some point.

![Diagram showing Games 0, 1, and 2 with $\Pi$, $\Pi_E$, and $B$.](image)
Bounding $B$:

Bad when have $(m, c)$ query to $E$ and $(\alpha, \beta)$ query to $\pi/\pi^{-1}$ such that:

\[ \alpha = m \oplus k \quad \beta = c \oplus k \]

Domain conflict \quad Range conflict

For a fixed $k$ indep of adv's view:

\[ \Pr[\text{one $E$ and $\pi/\pi^{-1}$ query is bad}] < \frac{2}{2^n}. \]

Total # of pairs is $Q_{enc} Q_{\pi}$, so by union bound

\[ \Pr[B] \leq \frac{2Q_{enc} Q_{\pi}}{2^n} \]

Then \[ |\Pr_1 - \Pr_2| \leq \Pr[B] \leq \frac{2Q_{enc} Q_{\pi}}{2^n} \]

So \[ \Pr_{Adv}[A^n, E^n] \leq \frac{2Q_{enc} Q_{\pi}}{2^n}. \]