April 12: Symmetric-Key Primitives

Review from Last Time

Definitions / Spinach

* Negligible functions, eff. algs, sec parameters
* PRG $\quad G: \{0,1\}^\lambda \to \{0,1\}^{\ell(\lambda)} \quad$ (for $\ell(\lambda) > \lambda$)

$$\{ s \xleftarrow{R} \{0,1\}^\lambda : G(s) \} \overset{c}{\approx} \{ z \leftarrow \{0,1\}^{\ell(\lambda)} \}$$

"stretch a short random seed into long random-looking string"

→ Blum Micali: PRG (one-bit PRG → poly(λ)-bit PRG)

→ Hybrid argument

* PRF $\quad F: \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$

$$\{ k \xleftarrow{R} \mathcal{K} : F(k, \cdot) \} \overset{c}{\approx} \{ f \xleftarrow{R} \text{Funs}[\mathcal{X}, \mathcal{Y}] : f(\cdot) \}$$

"Indist. from a random function"

* PRP $\quad F: \mathcal{K} \times \mathcal{X} \to \mathcal{X}$

$$\{ k \xleftarrow{R} \mathcal{K} : F(k, \cdot) \} \overset{c}{\approx} \{ f \xleftarrow{R} \text{Perms}[\mathcal{X}] : f(\cdot) \}$$

$$\begin{tikzcd}
\text{PRG} \arrow[r, bend left, "\text{GGM tree}"] & \text{PRF} \arrow[l, bend left, "\text{Counter mode}"] \arrow[r, bend left, "\text{Luby-Rackoff / 3-round Feistel}"] & \text{PRP} \arrow[l, bend left, "\text{Switching lemma}"]
\end{tikzcd}$$

If you have any, you have all (up to polynomial efficiency / security loss)

→ Questions?

One application of PRPs

"Format-preserving encryption"

encrypt $\{$credit card #s$\} \longrightarrow \{$credit card #s$\}$

Logistics - April 12

- HW0 due now
- Project proposal due next week
  - * ideas online
  - * talk to us at OH/after class
- Lecture scribe? (first/second half)
- Note cards

Cycle Walking

Per Kevin's Q:

I claimed that PRPs are useful for encrypting

$\{\text{credit card } \#\} \to \{\text{credit card } \#s\}$

but Feistel network only gives a PRP on n-bit strings.

Problem: The set $\mathcal{C}$ of valid cc #s is a smallish subset of $\{0,1\}^{64}$. Not all 16-digit #s are cc #s!

For example, cc #s have a "check digit"/CRC. We want a PRP on cc #s!

Say we have eff predicate

$$\text{Valid}: \{0,1\}^n \to \{0,1\}$$

Define $\mathcal{C} = \{ x \mid \text{Valid}(x) = 1 \} \subseteq \{0,1\}^n$

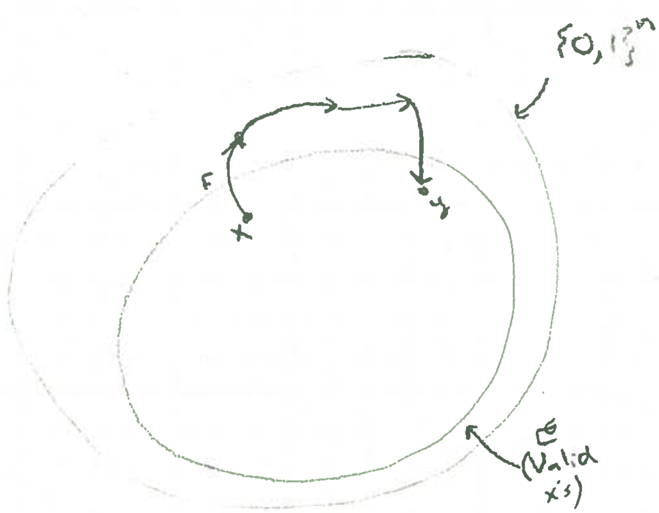
We have PRP $F: \mathcal{K} \times \{0,1\}^n \to \{0,1\}^n$. (PRP on strings)

We want PRP $F_c: \mathcal{K} \times \mathcal{C} \to \mathcal{C}$. (PRP on cc #s)

Claim (Shroeppel, Orman):

Given F, Valid, can construct F_c, as long as $\frac{|\mathcal{C}|}{2^n} > \frac{1}{\text{poly}(n)}$.

Pf Idea By picture



$F_c(k, x)$:

$y \leftarrow x$

do {

$y \leftarrow F(y)$

} while $\text{Valid}(y) \neq 1$

return y.

$F^{-1}(\cdot,\cdot)$ is similar.

Cycle Walking

Correctness: The expected # of invocations of F is $\frac{2^n}{|\mathcal{C}|} < \text{poly}(n)$.

As long as "valid" xs 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 $\pi \xleftarrow{\$} \text{Perms}[2^n]$.
2) Argue that cycle-walking using π gives a random permutation

$$\pi_c : \mathcal{C} \to \mathcal{C}$$

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, PRP, etc) requires $P \neq NP$ 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) — DES is a PRP
2) Change the model

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

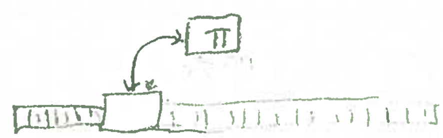
Now we will show how to use approach #2 to construct a PRP

Random Permutation Model

"Standard Model"

Both good guy and adv. use a Turing machine

"Random Permutation Model" (RPM)



* Both good guy and adv use TMs w/ oracle access to the same random public perm π, π^{-1}
* To compute $\pi(x)$ must query π

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

The catch:
1) π is exponentially large!
2) Schemes secure in RPM may be broken when π is instantiated with a real efficiently computable perm.

Still! * Useful analysis technique: If 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 <u>random function</u>
ideal cipher model " " " random family of πs.

<u>Even Mansour</u>

π is R.P., $\pi : \{0,1\}^n \to \{0,1\}^n$.

$E : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$

$$\left.\begin{array}{l} E(k, m) = k \oplus \pi(k \oplus m) \end{array}\right\}$$ Original paper uses two diff keys

$$D(k, c) = k \oplus \pi^{-1}(k \oplus c)$$
$$= k \oplus \pi^{-1}(k \oplus k \oplus \pi(k \oplus m))$$
$$= k \oplus \pi^{-1}(\pi(k \oplus m))$$
$$= k \oplus k \oplus m$$
$$= m$$

<u>Interesting</u> because...

* <u>It's</u> as simple as it gets!
* Basis for design of AES (SPN)
  ↳ AES128 uses 10-rounds of SPN
* Gives a "rigorous heuristic" for design of PRPs in practice

Even - Mansour

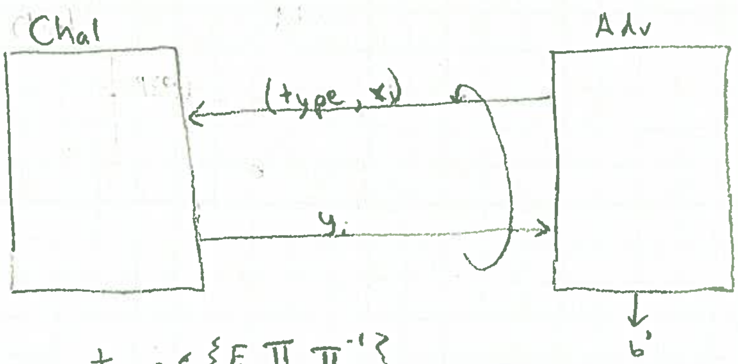
- Paper proves two non-standard sec properties
- Boneh-Shoup (~~follows~~ ~~K~~Kilian-Rogaway '00) show EM is a PRP.

Thm (Kilian, Rogaway) — (4.14 in Boneh-Shoup)
Let $A^{\pi, \pi^{-1}}$ be an adversary making at most Q_{enc} queries and at most Q_{π} permutation queries, then

$$\mathrm{PRPAdv}^{\pi}[EM, A] \leq \frac{2 \cdot Q_{enc} Q_{\pi}}{2^n}.$$

N.B. → We prove security against advs running in unbounded time.
→ Info-theoretic result.

Proof



type $\in \{E, \Pi, \Pi^{-1}\}$

Overview

Game 0: Challenger responds using EM cipher

Game 1: Rephrasing of game 0

Game 2: Challenger respond with independent permutation $\Pi_E(\cdot)$

Define $p_i = \Pr[\text{Adv outputs } 1 \text{ in game } i]$

Want to show $|p_0 - p_2| \le \frac{2 Q_{enc} Q_\pi}{2^n}$.

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

The challenger responds to E, Π, Π^{-1} queries

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

We represent Π as a set of (α, β) pairs

Proof

Game 0: (Real Construction)

setup: $\Pi \leftarrow \emptyset$

$k \xleftarrow{R} \{0,1\}^n$

E query on m:

$\alpha \leftarrow m \oplus k$

if $\Pi(\alpha)$ undefined

$\Pi(\alpha) \xleftarrow{R} \{0,1\}^n \setminus Range(\Pi)$

return $k \oplus \Pi(\alpha)$

Π query on α:

if $\Pi(\alpha)$ undef

$\Pi(\alpha) \xleftarrow{R} \{0,1\}^n \setminus Range(\Pi)$

return $\Pi(\alpha)$

Π^{-1} query on β:

if $\Pi^{-1}(\beta)$ undef:

$\Pi^{-1}(\beta) \leftarrow \{0,1\}^n \setminus Domain(\Pi)$

return α

Game 1 (Intermediate)

setup: $\Pi_E \leftarrow \emptyset$ ←----- Private random perm in PRP game

$\Pi \leftarrow \emptyset$ ←----- Public random perm

$k \xleftarrow{R} \{0,1\}^n$

E query on m

$\alpha \leftarrow m \oplus k$

if $\Pi_E(\alpha)$ $\boxed{\text{and } \Pi(\alpha)}$ undef

$\Pi_E(\alpha) \xleftarrow{R} \{0,1\}^n \setminus Range(\Pi_E)$

if $\Pi_E(\alpha) \in Range(\Pi)$

$\Pi_E(\alpha) \xleftarrow{R} \{0,1\}^n \setminus (Range(\Pi) \cup Range(\Pi_E))$

return $k \oplus \Pi_E(\alpha)$

Π query on α:

if $\Pi(\alpha)$ and $\Pi_E(\alpha)$ undef

$\Pi(\alpha) \xleftarrow{R} \{0,1\}^n \setminus Range(\Pi)$

if $\Pi(\alpha) \in Range(\Pi_E)$

$\Pi(\alpha) \xleftarrow{R} \{0,1\}^n \setminus (Range(\Pi) \cup Range(\Pi_E))$

return $\Pi(\alpha)$

Π^{-1} query on β:

if $\Pi^{-1}(\beta)$ and $\Pi_E^{-1}(\beta)$ undef

$\Pi^{-1}(\beta) \xleftarrow{R} \{0,1\}^n \setminus Domain(\Pi)$

if $\Pi^{-1}(\beta) \in Domain(\Pi_E)$

$\Pi^{-1}(\beta) \xleftarrow{R} \{0,1\}^n \setminus (Domain(\Pi) \cup Domain(\Pi_E))$

return $\Pi^{-1}(\beta)$

Proof (cont'd)

In Game 0: $E/\Pi/\Pi^{-1}$ queries give responses as in real construction

In Game 1: Everything is exactly as in game 0, except that we split Π into two parts: Π and Π_E.

$\to$ By construction $\Pi \cap \Pi_E = \emptyset$, so Π and Π_E together define a permutation

In Game 2: E queries responded to with Π_E

Π/Π^{-1} queries responded to with independent Π

$\Rightarrow E$ is a real random permutation

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

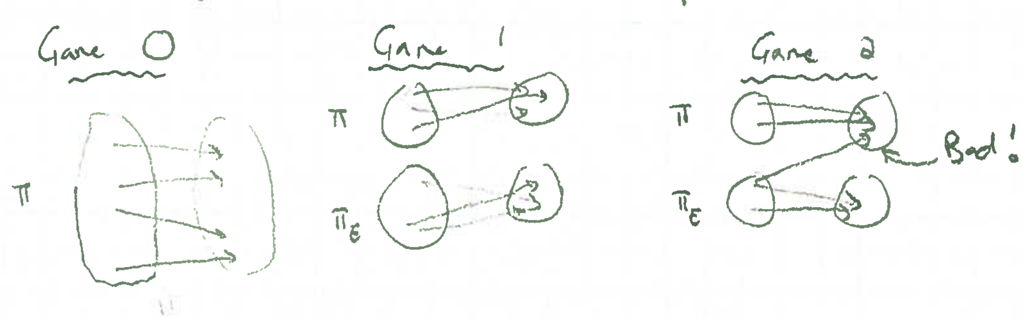
To complete proof, want to bound $|p_1 - p_2|$

Bad event B is in game 2 that

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

As long as $\overline{B}$, Games 1 and 2 are identical. (See Boneh-Shoup for details.)

Intuitively: Bad event is that Π and Π_E conflict — they give contradictory answers at some point.



Bounding B:

Bad when have (m, c) query to E and (α, β) query to π/π^{-1} such that.

$$\underbrace{\alpha = m \oplus k}_{\text{Domain conflict}} \qquad \underbrace{\beta = c \oplus k}_{\text{Range conflict}}$$

For a fixed k indep of adv's view

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

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

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

Then $|P_1 - P_2| \leq \Pr[B] \leq$

So $\text{PRPAdv}[A^{\pi}, E^{\pi}] \leq \dfrac{2 Q_{enc} Q_{\pi}}{2^n}$. ∎