Eliminating Random Permutation Oracles in the Even-Mansour Cipher

Zulfikar Ramzan

Joint work w/ Craig Gentry

DoCoMo Labs USA

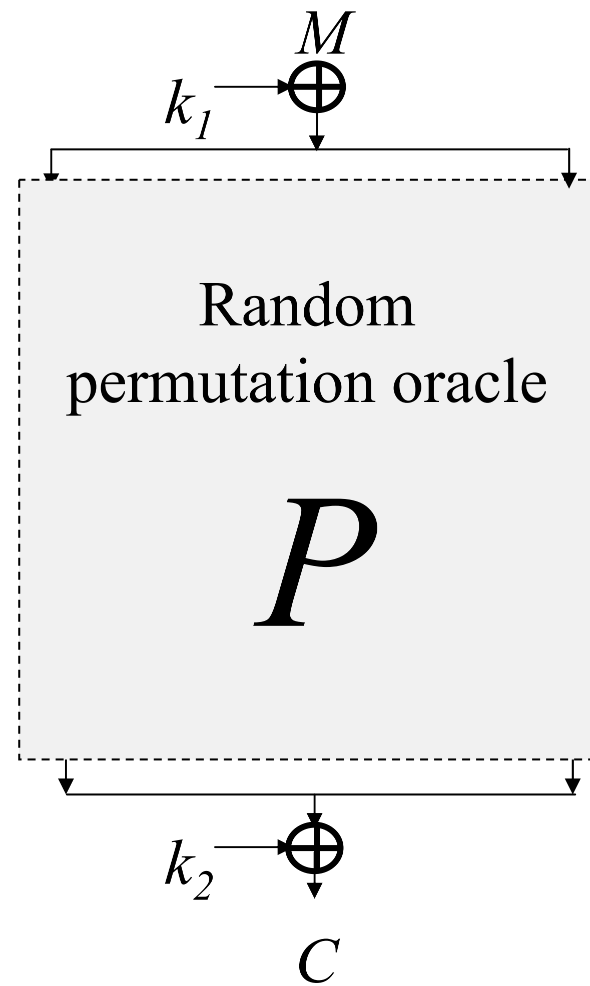
ASIACRYPT 2004

Outline

- Even-Mansour work and open problems.
- Main contributions (resolving open problems)
- Related work
- Formal security theorem & proof sketch
- Extensions & Negative results

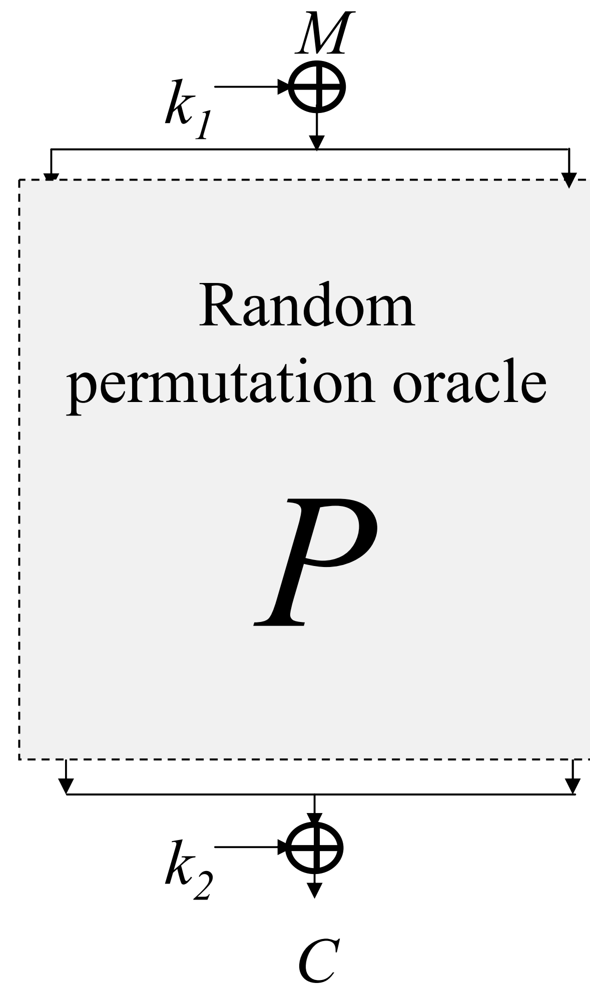
Even-Mansour Construction

- Goal: block cipher based on single (public) random permutation.
- C = k2 xor P(M xor k1)
- Security Model – Adversary:
  - makes chosen plaintext / ciphertext queries
  - has separate oracle access to P, P^{-1}.
- [EM91] proved: hard to invert (or compute forward direction of) cipher for un-queried plaintext/ciphertext pair.



Issues and Open Problems

- Security is proved in "Random *Permutation* Oracle Model."
  - How to instantiate Random Permutation Oracle?
- Security proved w.r.t. hardness of inversion / forgery.
  - But, there are stronger adversarial models.



Q1: Can we prove security outside random permutation oracle model?

Q2: Can we prove security w.r.t. to stronger adversarial model?

Our Contributions

Q1: Can we prove security outside the random *permutation* oracle model?

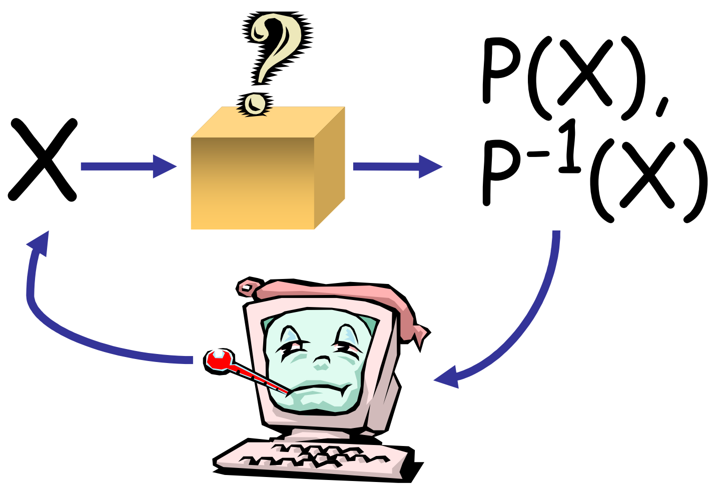
A1: Yes. We build the publicly-computable permutation using (publicly computable) functions. These functions are modeled as random *function* oracles; i.e., they're not necessarily *bijective*.

Q2: Can we prove security w.r.t. to stronger adversarial model?

A2: Yes. We prove super pseudorandomness (i.e., cipher is indistinguishable from a random permutation under chosen message/ciphertext attack).

Super Pseudorandom Permutations

- Block Cipher is super-pseudorandom if all Probabilistic Poly-time Turing Machines (PPTM) fail **Turing Style Test** of **Block Cipher vs. Truly Random Permutation**.



PPTM adaptively chooses plaintexts (resp. ciphertexts); is provided corresponding ciphertexts (resp. plaintexts).

Should be unable to distinguish cipher from truly random permutation on same domain

- Luby-Rackoff: constructed secure block cipher based on existence of one-way functions.

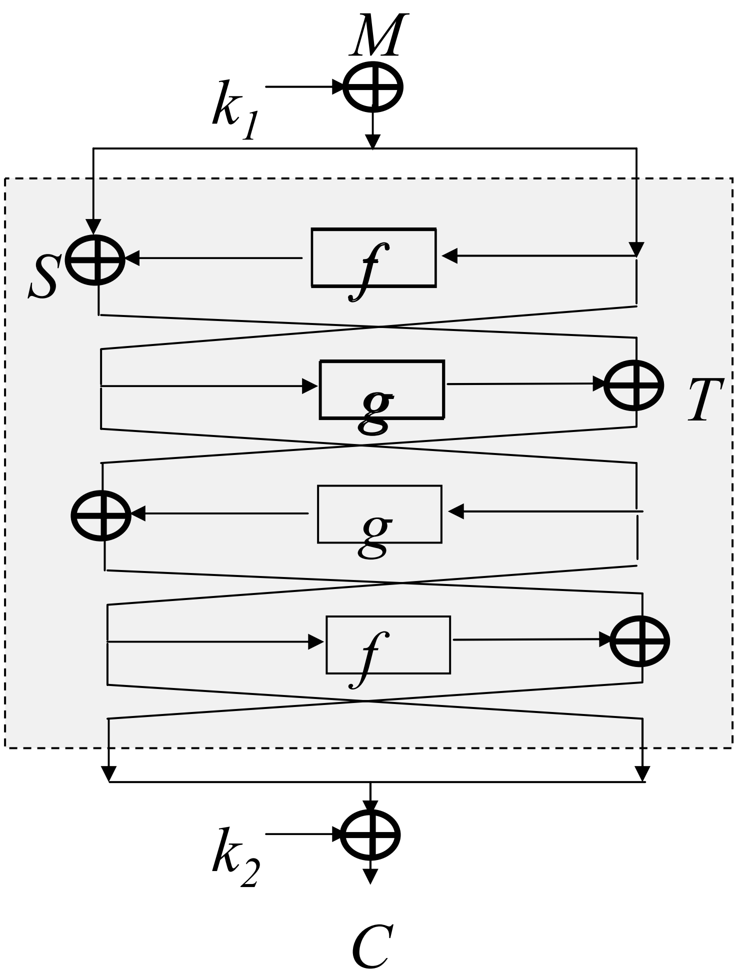
Health Warnings…

- Security in the random oracle model does not guarantee security in the real world [CGH97; MRH04; GTK03; BBP04]
- There are more efficient block cipher constructions in the random oracle model [Ramzan-Reyzin-2000].
- Our security analysis indicates that we need $2^{n/2}$ to be large where block size is 2n.

Main contribution: solve fundamental theoretical open problems of Even-Mansour work; we don't recommend this as a practical approach for building block ciphers.

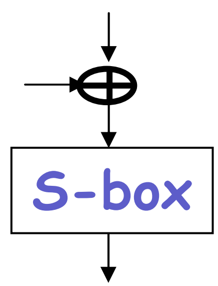
Our Construction

- Replace Random Permutation Oracle with Four Round Feistel.
- Round functions modeled as length-preserving random *function* oracles (note: may be non-injective).
- Our Results:
  - Instantiate (public) *permutation* using (publicly computable) random *function* oracles.
  - Prove *super-pseudorandomness*.
  - Therefore: *eliminated random permutation oracles* in Even-Mansour.
- **Note**: adversary has separate black-box access to ALL round functions.



Related Work: Luby-Rackoff

- LR88: 4-Round Feistel w/ keyed pseudorandom round functions => super pseudorandom permutation.
  - BUT: adversary not given separate access to internal round functions.
- LR88: originally motivated by security of DES.
  - Viewed their construction as "idealized" DES.
  - But, DES round functions (S-boxes) are keyed in simple way (i.e., XOR key with input before applying S-box)
  - LR88 uses pseudorandom round functions (which don't involve simple keying…)



We consider "simple" keying; so, our model is arguably a more apt idealization.

Related Work Continued

- Ramzan-Reyzin Round Security Framework:
  - Allows adversaries access to internal rounds.
  - We can phrase security theorems using round security language.
  - There are similarities, but Ramzan-Reyzin constructs still had some keyed functions not accessible to adversary.
  - In this work: (essentially) no keyed functions. All funcs are separately accessible to adversary.
  - The respective proof strategies have some subtle differences (e.g., we need an extra hybrid).

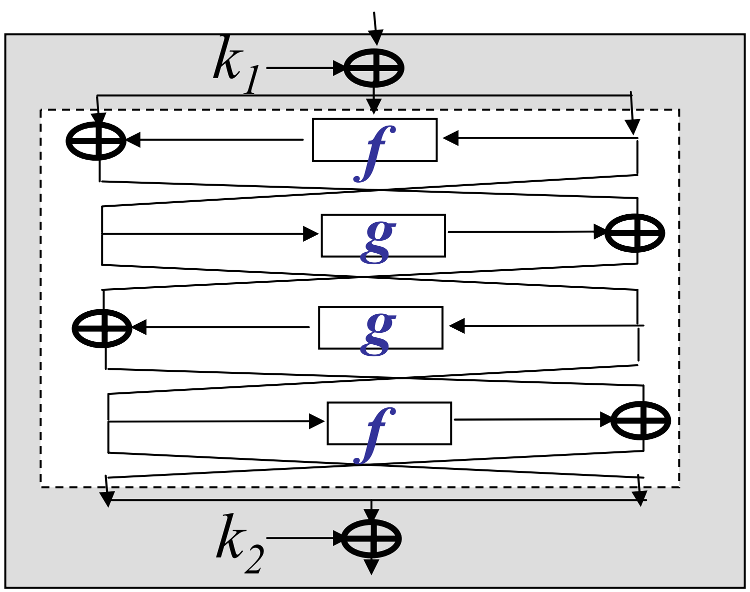
Two Worlds - Adversarial Model

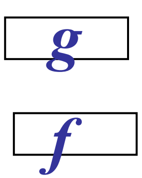
World 1: black-box oracles for

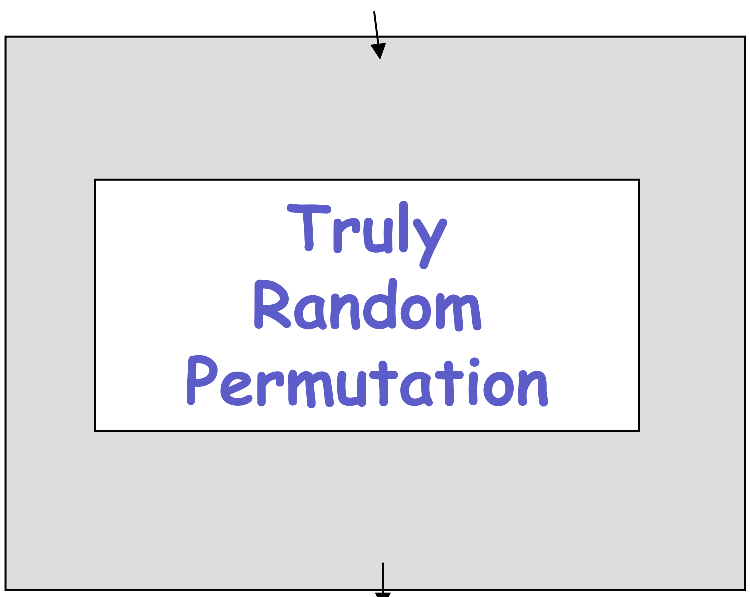
- forward + reverse direction of cipher.
- round functions inside cipher (both modeled as random function oracles)

World 2: black-box oracles for

- forward + reverse direction of truly random permutation.
- two random oracles

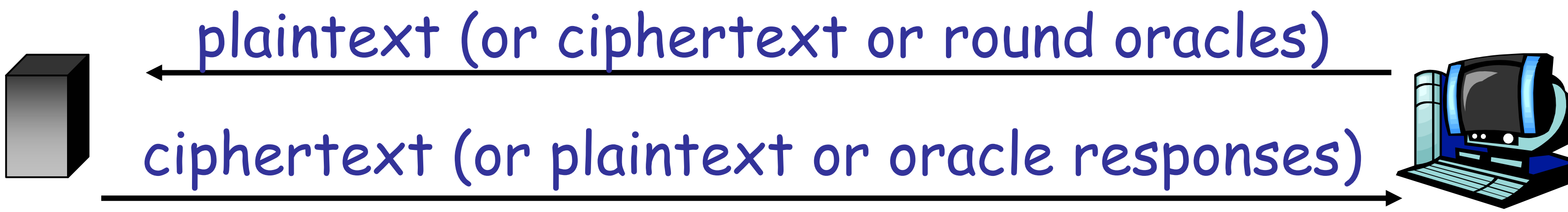






Theorem Statement: Adversarial Model

Adversary A is put in one of the two worlds; he makes q queries total to his three black boxes



Theorem: A successfully distinguishes world one from world two with advantage at most:

$$O(q^2 * 2^{-n}),$$

where block size is $2n$.

Proof Ideas… 1 – General Scheme

- Identify "BAD" conditions (as function of keys)
- Show: If for specific pair of keys, BAD conditions don't happen, then
  - Adversary's transcript view of interacting with World 1 (our construction) is distributed identically to…
  - Adversary's transcript view of interacting with World 2 (truly random permutation)…
- Show: Bad conditions happen with probability $O(q^2 * 2^{-n})$,

For technical reasons, we must compose the above paradigm with itself, considering two classes of bad conditions, and we need an additional hybrid in between.

- Finally, we apply "probability argument" to above

Proof Ideas… 2 – "Probability Argument"

- First, express adversary's (in)ability to distinguish between worlds in terms of statistical distance between transcripts (Apply Triangle Inequality several times…)
- Re-express probabilities to be conditioned on whether BAD events occur. (Apply Triangle inequality several more times…)
- Manipulate formulas to show that adversary's advantage is bounded by probability of BAD conditions occurring.

Proof Ideas… 3 – Actual BAD conditions

BAD conditions depend on possible transcript and probability of BAD occurring is taken over choice of key.

- Inputs to f (resp. g) during query to block cipher black box matches input to f (resp. g) during query to random oracle.
- Inputs to f (resp g) during different block cipher queries match.

If BAD doesn't happen:

1) external oracles don't see same inputs as internal oracles, so they are useless.

2) All outputs from cipher are uniformly distributed.

Intuition: BAD conditions unlikely since randomly chosen key directly or indirectly masks function inputs => collisions unlikely

Extensions: Recycling Key Material

- Proof only requires key to be XOR'ed into left half of input and right half of output.
  - o Immediate 2x reduction in key material.
- Q: Can we go further? i.e., use same key at beginning and end??
  - o XOR is symmetric;
  - o same key used at beginning and end is even more symmetric!
  - o The construction would behaves like an involution (not very random)!
- But, using observation from [PRS02] : if we use group operations other than XOR (i.e., where $a+a \neq 0$), then we can recycle keys.

Negative Results…

- Can recover entire 4n bit key with $2^{n+0.5}$ known plaintexts and $2^{n+0.5}$ work.
  - Basic application of the "Sliding with a Twist" attack [BW00].
  - The attack doesn't really exploit Feistel structure.
- Can attack 3 Feistel round version of our scheme
  - Straightforward adaptation of attack on 3-round Luby-Rackoff ciphers

Open Area: There's a gap between lower bounds from best known attacks and upper bounds from security analysis.

Conclusions

- Resolved fundamental open questions Mansour work.
  - Demonstrated that underlying random permutation oracle could be instantiated with construction involving random function oracles.
- We also better model idealized DES-like ciphers, which was a motivating goal for the Luby-Rackoff work.
- Open problem: decrease the gap between best known attacks and security analysis.

Thank You! Questions?