From Distinguishers to Key Recovery: Improved Related-Key Attacks on Even-Mansour

Pierre Karpman

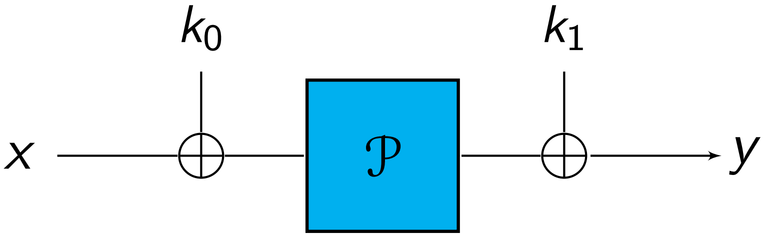
Inria and École polytechnique, France
Nanyang Technological University, Singapore

ISC, Trondheim, 63° N
2015–09–09

Even-Mansour block ciphers

- How to construct a block cipher easily from a public permutation $\mathcal{P}$?
- Simple: $\mathcal{E}((k_1, k_0), x) := \mathcal{P}(x \oplus k_0) \oplus k_1$
- For an n-bit block, proba. of recovering the key with time T and data D is $\leq \mathcal{O}(DT \cdot 2^{-n})$ (Even & Mansour, 1991)
- $\Longrightarrow \mathcal{O}(2^{\frac{n}{2}})$ security

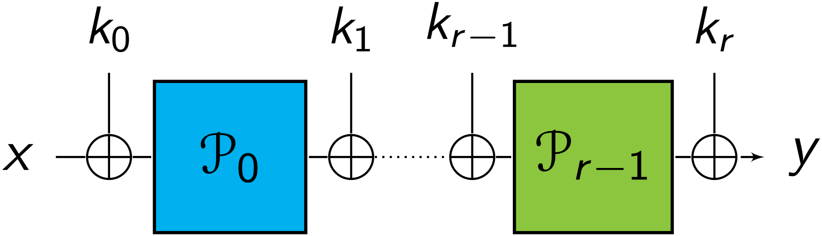
Even-Mansour



Improvement: Iterated Even-Mansour

- Get better security by iterating $\mathcal{P}$s
- $\mathsf{IEM}^r((k_r, k_{r-1}, \ldots, k_0), p) :=$
  $\mathcal{P}_{r-1}(\mathcal{P}_{r-2}(\ldots \mathcal{P}_0(p \oplus k_0) \oplus k_1) \ldots) \oplus k_r$
- $\mathcal{O}(2^{\frac{rn}{r+1}})$ security (Chen & Steinberger, 2014)

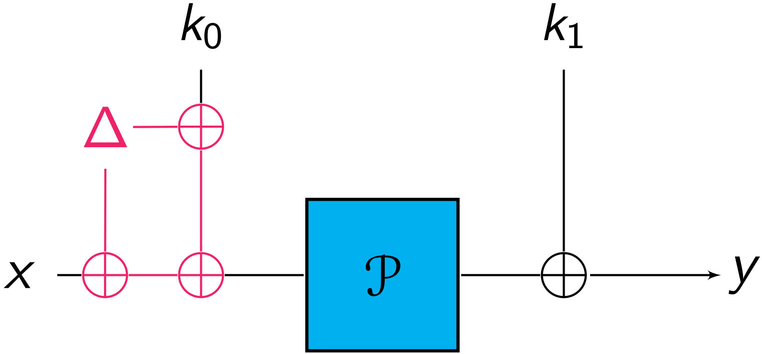
IEM not REM



But what happens with related-keys?

- There is a trivial RK distinguisher for Even-Mansour
- $\mathcal{E}((k_1, k_0 \oplus \Delta), x \oplus \Delta) = \mathcal{E}((k_1, k_0), x)$ for $\mathcal{E}$ an Even-Mansour cipher
- Also works for IEM with independent keys

RK Distinguisher



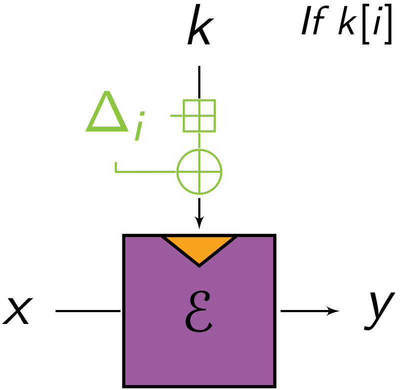
Provable bounds for related-keys

- Some variants of EM resist to RKA (Cogliati & Seurin, 2015), (Farshim & Procter, 2015)
- $\Rightarrow$ IEM with one key and at least 3 rounds
- $\Rightarrow$ IEM with one key with a non-linear key schedule
- $\mathcal{O}(2^{\frac{n}{2}})$ security for both (can't do better)

RKA models

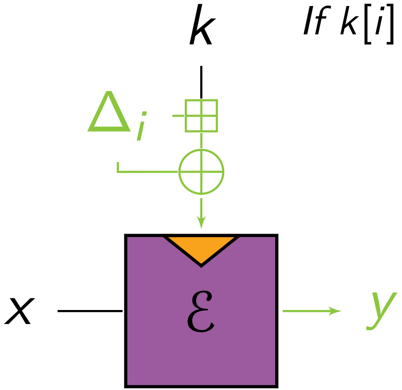
- Not all related-key attacks make sense (Bellare & Kohno, 2003)
- Queries to both $(k \oplus \Delta)$, $(k \boxplus \Delta)$ trivially break most ciphers

Trivial RKA illustrated (vol. 1)



If $k[i] = 0$

Trivial RKA illustrated (vol. 2)



If $k[i] = 1$

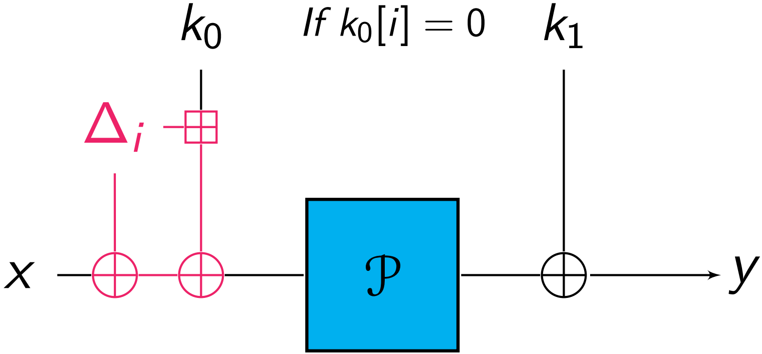
RKA models (cont.)

- Some RKA models still make sense
- Accessing only $(k \oplus \Delta)$ is sound
- Accessing only $(k \boxplus \Delta)$ is sound
- $\Longrightarrow$ A cipher resistant to RKA should resist to both classes

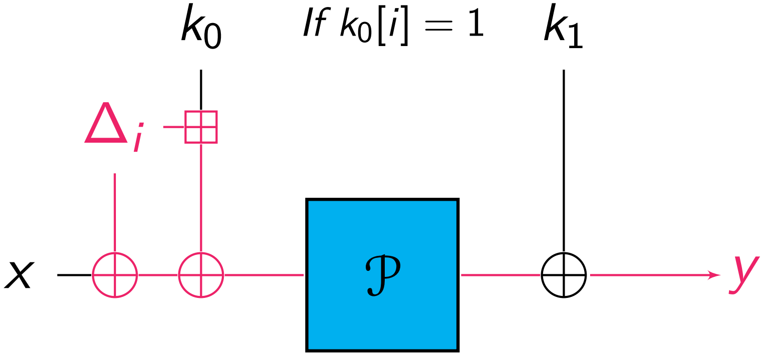
Back to Even-Mansour

- The RK distinguishers can be converted to key recovery
- Switch to queries to $(k \boxplus \Delta)$
- (Still only one RK class)

Even-Mansour RK key recovery (k[i] = 0)



Even-Mansour RK key recovery (k[i] = 1)



Even-Mansour RK key recovery (summary)

- Recover the key with linear complexity(!!)
- Only works on "distinguishable" constructions
- $\Longrightarrow$ (1,2)-round (I)EM, n-round IEM with independent keys

Let's break stuff!

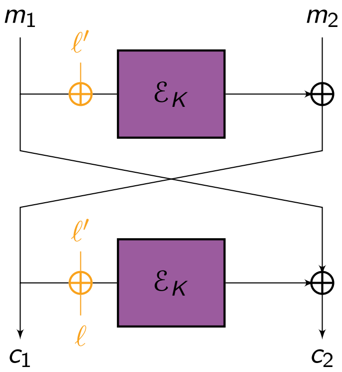
Application: RKA on Prøst-OTR

- Prøst: a permutation
- ⇒ "Prøst/SEM": an EM cipher with Prøst
- OTR: an AE mode
- ⇒ Prøst-OTR: an instantiation of OTR with Prøst/SEM
- Prøst-{COPA,OTR,APE}: 1st-round candidate to CAESAR (not selected for round 2)

Objective: key-recovery

- Apply the RKA on EM to Prøst/SEM embedded into OTR
- (Needs some adapting)
- (Direct application to Prøst/SEM also works but doesn't mean much)

Encryption with OTR



Encryption with Prøst-OTR (first block)

- $c_1 = \mathcal{F}(k, n, m_1, m_2) = \mathcal{E}(k, \ell'(k, n) \oplus m_1) \oplus m_2$
- $k, m_1, m_2, \ell \in \{0,1\}^\kappa,\ n \in \{0,1\}^{\frac{\kappa}{2}}$
- $\ell = \mathcal{E}(k, n||10^*)$
- $\ell' = 4 \otimes \ell$ (in $\mathbb{F}_{2^\kappa}$) $(4 := x^2)$

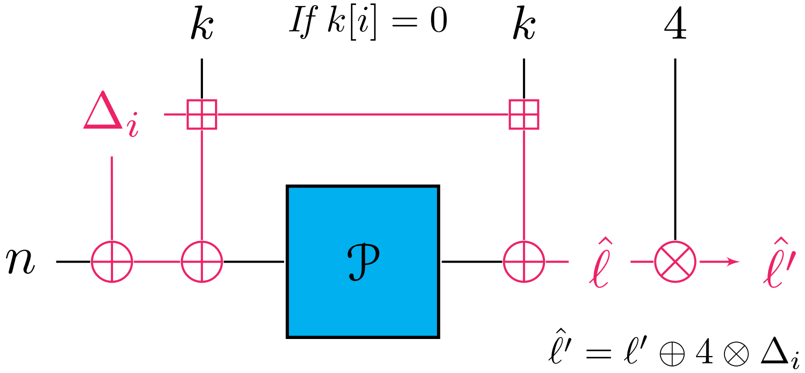
RK key recovery (upper half)

- Query an RK encryption oracle for $\mathcal{F}$
- Deduce key bits one by one (two queries per bit)

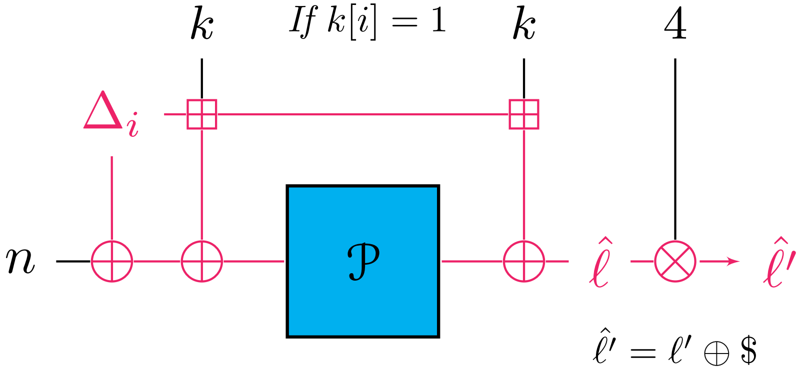
1. $c_1 := \mathcal{F}(k, n, m_1, m_2)$
2. $\hat{c}_1 := \mathcal{F}(k \boxplus \Delta_i, n \oplus \Delta_i, m_1 \oplus \Delta_i \oplus 4 \otimes \Delta_i, m_2)$
3. $\mathrm{k[i]} = 0$ iff $c_1 \oplus \hat{c}_1 = \Delta_i$

$\Longrightarrow$ Must add Δ_i to the nonce: only works if $i \geq \frac{\kappa}{2}$

Why this works (computation of ℓ', k[i] = 0)



Why this works (computation of ℓ', k[i] = 1)



Finishing up (k[i] = 0)

- $c_1 = \mathcal{E}(k, \ell' \oplus m_1) \oplus m_2$
- $\hat{c}_1 = \mathcal{E}(k \boxplus \Delta_i, \hat{\ell}' \oplus m_1 \oplus \Delta_i \oplus 4 \otimes \Delta_i) \oplus m_2$
- $\hat{c}_1 = \mathcal{E}(k \boxplus \Delta_i, \ell' \oplus 4 \otimes \Delta_i \oplus m_1 \oplus \Delta_i \oplus 4 \otimes \Delta_i \oplus m_2$
- $\hat{c}_1 = \mathcal{E}(k \boxplus \Delta_i, \ell' \oplus m_1 \oplus \Delta_i) \oplus m_2$
- $\hat{c}_1 = c_1 \oplus \Delta_i$

RK key recovery (lower half)

- Can't add Δ_i in the nonce ($i < \frac{\kappa}{2}$)
- Solution:
  - Zero the known part of the key
  - Use $\boxplus/\boxminus$ to propagate the difference (Δ_i) up
  - Cancel it with $\Delta_{\kappa/2}$ in the nonce
- No details here (not hard, but a bit ugly)
- BTW, wouldn't have worked with a padding $0^*1||n$

Code of the attack (upper half)

```
uint64_t recover_hi(uint64_t secret_key)
{
uint64_t kk = 0;
for (int i = 62; i >= 32; i--)
{
uint64_t m1, m2, c11, c12, n;

m1 = (((uint64_t)arc4random()) << 32) ^ arc4random();
m2 = (((uint64_t)arc4random()) << 32) ^ arc4random();
n  = (((uint64_t)arc4random()) << 32) ^ 0x80000000ULL;
c11 = potr_1(secret_key, n, m1, m2);
c12 = potr_1(secret_key + DELTA(i), n ^ DELTA(i), m1 ^ DELTA
    (i) ^ TIMES4(DELTA(i)), m2);

if (c11 != (c12 ^ DELTA(i)))
kk |= DELTA(i);
}
return kk;
}
```

Code of the attack (lower half)

```
uint64_t recover_lo(uint64_t secret_key, uint64_t hi_key)
{
uint64_t kk = hi_key;
for (int i = 31; i >= 0; i--)
{
uint64_t m1, m2, c11, c12, n;
uint64_t delta_p, delta_m;
m1 = (((uint64_t)arc4random()) << 32) ^ arc4random();
m2 = (((uint64_t)arc4random()) << 32) ^ arc4random();
n  = (((uint64_t)arc4random()) << 32) ^ 0x80000000ULL;
delta_p = DELTA(i) - MSB(kk) + (((LSB(~kk)) >> (i + 1)) << (
    i + 1));
delta_m = DELTA(i) + MSB(kk) + LSB(kk);
c11 = potr_1(secret_key + delta_p, n ^ DELTA(32), m1 ^ DELTA
    (32), m2);
c12 = potr_1(secret_key - delta_m, n, m1 ^ TIMES4(DELTA(32))
    , m2);
if (c11 == (c12 ^ DELTA(32)))
kk |= DELTA(i);
}
return kk;
}
```

Other targets?

- Not many vulnerable EM in the wild...
- LED, Minalpher resist RK distinguishers (hence key recovery)
- Applicable to PRINCE, PRIDE (but they don't claim RK resistance)

Lesson to learn

Allowing RK distinguishers on EM $\equiv$ allowing (linear-time) RK key recovery $\mod$ change of RK class