The Lightweight Block Cipher LED as an Inspiration for Cryptanalysis

Itai Dinur
École normale supérieure, France
Block Ciphers

- A collection of permutations over $n$-bit strings indexed by a secret key $K$
Block Ciphers

• Block ciphers are traditionally designed with complex key schedules

• Ensures fast diffusion of key bits into the state
  • Provides enhanced resistance against cryptanalytic attacks such as meet-in-the-middle
Block Ciphers

• In recent years, with the rise of lightweight cryptography, designers are forced to simplify the key schedule.

• There is renewed interest in the Even-Mansour scheme - A simple construction of a block cipher proposed in 1991.
The Even-Mansour Scheme (1991)

- A simple construction of a block cipher using 2 keys of \(n\) bits and a public permutation \(F\)

- **Information-theoretic** security lower bound:
  - Assume that \(F\) is *randomly chosen*
  - Assume that we obtain \(D\) plaintext-ciphertext pairs \((P_i, C_i)\)
  - Then, any successful key-recovery attack that evaluates \(F\) on \(T\) inputs \(X\) must satisfy \(T \geq 2^n/D\)
The SlideX Attack [DKS ‘12]

- Security: $TD=2^n$ using the SlideX attack (DKS, Eurocrypt ‘12)

- Given $D=2^{n/2}$ the scheme can be broken in $T=2^{n/2}$
  - Considering $D>2^{n/2}$ is less interesting
The Iterated EM Scheme

- EM-based schemes are a very hot research area
  - Over 10 papers in major crypto conferences since 2011

- There are many possible key schedules
The Lightweight Block Cipher LED

• LED is a 64-bit lightweight block cipher presented at CHES 2011 by Guo et al.

• Two main versions: LED-64 and LED-128
 LED-64

- LED-64 is an 8-round EM scheme with 1 key
LED-128

- LED-128 uses 2 alternating keys and has 12 rounds
Summary

- Study the security of LED as an iterated Even-Mansour scheme

- The cryptanalytic techniques do not exploit the properties of the internal permutations of LED
  - Applicable to any block cipher that uses a similar key schedule
SlideX on EM with 1 Key [DKS ‘12]

\[ P_i + K = X_i \text{ and } C_i + K = Y_i \Rightarrow P_i + C_i = X_i + Y_i \]

- For each \((P_i, C_i)\):
  - Calculate \(P_i + C_i\) and store it in a sorted table next to \(P_i\)

- For arbitrary values \(X_j\):
  - Calculate \(Y_j = F(X_j)\) and search \(X_j + Y_j\) in the table
  - For each match, test the suggestion for \(K = P_i + X_j\)
The attack succeeds once we have \((P_i, X_j)\) such that \(K = P_i + X_j\).

In order to obtain such a pair \(w.h.p\) we need a total of about \(2^n\) pairs \((P_i, X_j)\), i.e. \(TD = 2^n\).

For an arbitrary \(X_j\) we expect less than one match in the table.

- The time complexity of the attack is \(\max(T, D) = T\) (assuming \(D \leq 2^{n/2}\)).
- The memory complexity \(M\) of the attack is equal to \(D\).
2-Round Iterated EM with 1 Key

• Does not provide $n$-bit security as shown at FSE 2013 [NWW ‘13]
A Variant of the Previous Attack

[NWW ‘13] – Main Idea

- \( P_i + V_i = X_i + Y_i \rightarrow X_1 + Y_1 = X_2 + Y_2 = \ldots = X_t + Y_t = \Delta \) then \( P_1 + V_1 = P_2 + V_2 = \ldots = P_t + V_t = \Delta \)

- A \( t \)-way collision on the public \( F'_1(X) = X + F_1(X) \) gives a \( t \)-way collision on \( P_i + V_i \) with the same value \( \Delta \)

- Given \( \Delta \), and a random \( P_i \), then \( V_i = P_i + \Delta \) with probability \( t/2^n > 1/2^n \)
A Variant of the Previous Attack
[NWW ‘13]

- **Preprocessing**: Evaluate $F_1$ on arbitrary inputs $X$, find a $t$-way collision on $F'_1(X) = X + F_1(X)$ and denote the colliding value by $\Delta$

- **Online**: For each $(P_i, C_i)$:
  - Assume that $V_i = P_i + \Delta$ and compute $W_i = F_2(V_i)$
  - Compute a suggestion for $K = W_i + C_i$ and test it

![Diagram]

\[ P_i \xrightarrow{+\Delta} X_i \xrightarrow{F_1} Y_i \xrightarrow{V_i} F_2 \xrightarrow{W_i} C_i \]
A Variant of the Previous Attack [NWW ‘13] - Analysis

- The data complexity is $D = 2^n / t$
  - in order to find a $P_i$ such that $V_i = P_i + \Delta$ and recover $K$
- The **online** time complexity is also $2^n / t$
- What is the complexity of the preprocessing?
A Variant of the Previous Attack
[NWW ‘13] - Analysis

- If we evaluate $F'_1$ on all $2^n$ inputs, the attack will not be faster than exhaustive search
- We evaluate $F'_1$ on a $\lambda < 1$ fraction of the inputs
- The preprocessing time complexity is $\lambda 2^n$
  - in which we find a $t$-way collision
A Variant of the Previous Attack
[NWW ‘13] - Analysis

• The **total** time complexity is $\lambda 2^n + 2^n / t$
• To calculate the **optimal** time complexity, we need to understand the **tradeoff** between $\lambda$ and $t$
• What is the largest $t$-way collision we expect when evaluating a $\lambda$ fraction of inputs for $F'_1$?

\[
P_i \xrightarrow{K} X_i \xrightarrow{F_1} Y_i \xrightarrow{K} V_i \xrightarrow{F_2} W_i \xrightarrow{K} C_i
\]
A Variant of the Previous Attack
[NWW ‘13] - Analysis

- $F'_1(X) = X + F_1(X)$ is a function from $n$ bits to $n$ bits
- A function can be described using a bipartite graph, mapping inputs to outputs
A Variant of the Previous Attack
[NWW ‘13] - Analysis

- Assuming $F_1$ is a random permutation, $F'_1(X)=X+F_1(X)$ is (very close to) a random function mapping $n$ bits to $n$ bits.
- The in-degree of a vertex in the range of $F'_1$ is distributed according to the Poisson distribution.
  - The expectation equal to the average in-degree $\lambda$.
A Variant of the Previous Attack
[NWW ‘13] - Analysis

• We expect \( \frac{2^n \lambda^t e^{-\lambda}}{t!} \) vertices with an in-degree of \( t \) for \( F'_1 \)
A Variant of the Previous Attack

[NWW ‘13] - Analysis

- The tradeoff between $\lambda$ and $t$ is enforced by $(2^n\lambda^t e^{-\lambda})/t! \geq 1$
- Taking $\lambda \approx 1/n$ gives $t \approx 1/\lambda \approx n$ and minimizes $T \approx 2^n/n$
  - This is faster than exhaustive search by a factor of about $n$, which grows to infinity with $n$
- For $n=64 \rightarrow T \approx 2^{64}/64 \approx 2^{60}$ and also $D \approx 2^{60}$, $M \approx 2^{60}$
A First Optimization: Reducing the Data Complexity – Main Idea

- Once we take $\lambda$ and $t$ for which $(2^n\lambda^t e^{-\lambda})/t! \geq 1$, and slightly reduce $t$, the number of $t$-way collisions grows **rapidly**
- For $n=64$ and $2^{60}$ inputs we expect:
  - 4 10-way collisions
  - 95 9-way collisions
  - Over 100,000 8-way collisions
A First Optimization: Reducing the Data Complexity

- **Preprocessing**: Evaluate $F_1$ on arbitrary inputs $X$, find $t_1, t_2, \ldots, t_j$-way collisions on $F'_1(X) = X + F_1(X)$ and denote the colliding values by $\Delta_1, \Delta_2, \ldots, \Delta_j$

- **Online**: For each $(P_i, C_i)$:
  - For each $\Delta_j$:
    - Assume that $V_i = P_i + \Delta_j$ and compute $W_i = F_2(V_i)$
    - Compute a suggestion for $K = W_i + C_i$ and test it

![Diagram](image-url)
A First Optimization: Reducing the Data Complexity – Analysis

• How much data do we need?
• Let $\xi = (t_1 + t_2 + \ldots + t_I) / I$
• The collisions on $\Delta_1, \Delta_2, \ldots, \Delta_I$ “cover” $I\xi$ values of $X_i$ for which we need to find a matching $P_i$
• By the birthday paradox we need only
  $D = 2^n / I\xi \approx 2^n / \max(t_i) / I$
• For $n=64$ we greatly reduce the data complexity from $2^{60}$ to $2^{45}$ by taking $\xi = 8$ rather than $\max(t_i) = 10$
  • The time and memory complexities slightly increase but remain about $2^{60}$
3-Round Iterated EM with 1 Key

- The attack on 2-round EM was already somewhat marginal
- 3-round EM **does not** provide n-bit security as well!
The Main Idea of the Attack

• We know how to predict $W_i$ with a higher probability than a random guess
• Given $W_i$ and $C_i$ we remain with a 1-round EM with 1 key and can apply the SlideX attack
The Basic 3-Round Attack

- **Preprocessing**: Find a $t$-way collision on $F'_1(X)$ and denote the colliding value by $\Delta$
- Evaluate $F_3$ on inputs $U$ and store $U+Z$ in a sorted table
- **Online**: For each $(P_i, C_i)$:
  - Assume that $V_i = P_i + \Delta$ and compute $W_i = F_2(V_i)$
  - Search $W_i + C_i$ in the table
  - For each match obtain $Z_i$ and test $K = Z_i + C_i$
The Basic 3-Round Attack - Analysis

- For each \((P_i, C_i)\) the probability that \(W_i = F_2(P_i + \Delta)\) is \(t/2^n\)
- We expect to correctly calculate \(W_i\) for \(Dt/2^n\) plaintexts
- The table of \(F_3\) needs to have \(2^{2n}/Dt\) entries
- The time complexity of the attack is \(\lambda 2^n + (1/D)(2^{2n}/t) + D\)
- The optimal attack is obtained for \(D \approx 2^n/\sqrt{t}\), which gives \(T \approx 2^n/\sqrt{n}\)
  - Faster than exhaustive search only by a factor of \(\sqrt{n}\)
Optimizing the 3-Round Attack

• Apply the same optimization as in the 2-round attack to reduce the data complexity

• Use the freedom to choose the inputs on which we evaluate $F_1$ and $F_3$ in order to reduce the time complexity
Optimizing the 3-Round Attack

- $P_i + C_i = X_i + Z_i$ and in particular, this holds if we only consider the $m$ most significant bits
- Evaluate $F_1$ on inputs $X$ where the $m$ MSBs are 0
- Evaluate $(F_3)^{-1}$ on inputs $Z$ where the $m$ MSBs are 0
- For $(P_i, C_i)$ if we evaluated $F_1$ on $X_i$ and we evaluated $(F_3)^{-1}$ on $Z_i$ then the $m$ MSBs of $P_i$ and $C_i$ must be equal!
- This allows us to immediately filter most $(P_i, C_i)$
The Optimized 3-Round Attack

- The optimization gives us $T \approx 2^n/n$

- This is about the same time complexity as the 2-round attack!
Application to LED-64

- LED-64 is an 8-round EM scheme with 1 key

- We can directly apply our attack to 3-round LED-64 with $T \approx 2^{60}$, $M \approx 2^{60}$ and $D = 2^{49}$
Application to LED-128

• **LED-128** uses 2 alternating keys and has **12** rounds

• We use the new techniques to attack **8** rounds!
Application to LED-128

- Guess $K_1$ in an outer loop
- We remain with a 3-round EM scheme with 1 key
- We obtain $T \approx 2^{124}$, $M \approx 2^{60}$ and $D = 2^{49}$
Involution

- In practice the permutations $F_i$ can be constructed using a block cipher without the key schedule.
- Many of these constructions have the property that they are equal to their inverses.
- A permutation $F$ is called an involution if $F = F^{-1}$.
Fixed Points of Involutions

- A random involution has an expected number of $2^{n/2}$ fixed-points.

- $x = F(x) \rightarrow F'(x) = x + F(x) = 0 \rightarrow$ the vertex 0 in the graph of $F'(x)$ has an expected in-degree of $2^{n/2}$.
  - This is much larger than the $O(n)$ in-degree of a vertex of maximal in-degree, when $F$ is a random permutation.
Applications to Iterated EM

- A 2-round iterated EM scheme with 1 key can be attacked in $T \approx 2^n/t$
- When $F_1$ and $F_2$ are random permutations $T \approx 2^n/n$
- When $F_1$ (or $F_2$) is a random involution $T \approx 2^{n/2}$
  - The memory and data complexities are also significantly reduced
Applications to Iterated EM

- A 3-round iterated EM scheme with 1 key can be attacked in $T \approx 2^n/\sqrt{vt}$
- When all permutations are random $T \approx 2^n/\sqrt{n}$
- When $F_1$ (or $F_2$ or $F_3$) is a random involution $T \approx 2^{3n/4}$
  - The memory and data complexities are also significantly reduced

![Diagram of iterated EM scheme](image-url)
A Surprising Application

• A 2-round iterated EM scheme with 1 key with random permutations can be attacked in $T \approx 2^{n/n}$
• Add an arbitrary involutinal round (unrelated to the original permutations)
• This significantly reduces the security to $T \approx 2^{3n/4}$ !!
  • Also significantly reduces the data and memory complexities of the attack
Conclusions

- We presented attacks on several schemes based on iterated Even-Mansour
- We attacked 3 out of 8 rounds of LED-64
- We attacked 8 out of 12 rounds of LED-128
- The attacks can be applied to several other block ciphers such as Zorro and AES²
- The attacks are unlikely to be practically significant
- They show that a 1-key EM scheme needs to have at least 4 rounds to provide n-bit security
Thank you for your attention!