Linear Cryptanalysis

History

- Matsui
- Originally developed to analyse the block cipher FEAL
- Formalized and applied to DES in 1993
- Known plaintext attack
- Breaks DES with 2^{43} known plaintexts
- First attack on DES that was really implemented
- Apparently not known to the DES designers

Linear approximations and bias

- Boolean function f(x)
- Linear function U(x) (*approximation*)
- Bias ε = Prob(f(x) = U(x)) – ½
- $-½ \le \varepsilon \le ½$
- Bias = 0 indicates bad approximation
- Bias = ½ indicates good approximation
  - -½ is equally useful

Other notations

- x is a bit vector (column)
- $U(x) = u^T x$ where u is a column vector
- x can also be mapped to element of $GF(2^n)$
- U(x) = Trace(u x) where $u \in GF(2^n)$
  - Interesting notation if the cipher is defined over $GF(2^n)$

Linear approximations of S-boxes

- Vector Boolean function (S-box) S(x)
- Linear functions U(x) en V(S(x))
- Bias ε = Pr(V(S(x)) = U(x)) – ½
- DES designers made sure that the S-box outputs are not close to linear (i.e. have small ε)
- But they forgot about linear combinations of the S-box outputs
  - This fact was noticed almost immediately,
  - But it was not known how to exploit it

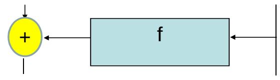
Approximations of linear functions

- Vector notation: y = A x
- $Pr(v^T y = u^T x) = ?$
  - $v^T y = v^T (A x) = (A^T v)^T x$
  - $\Rightarrow v^T y = u^T x$ if and only if $A^T v = u$
- A linear function has exactly one approximation which holds with probability 1
- All other approximations hold with probability 1/2

Finding good approximations

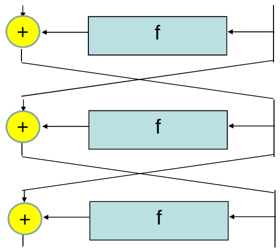
- For linear maps: deterministic relation
- For S-boxes: try all possibilities
  - Walsh-Hadamard transform
- For (several rounds of) a block cipher: approximate individual components and combine

Example: 1-Round DES



- S-box 5: $x_1 = y_1 + y_2 + y_3 + y_4$ with prob. 12/64
- f: $x_{16} = y_3 + y_8 + y_{14} + y_{25}$ with prob. 12/64
- 1 round: $x_{48} = x_3 + y_3 + x_8 + y_8 + x_{14} + y_{14} + x_{25} + y_{25}$ with prob. 12/64

Example: 3-Round DES



$P_3 + P_8 + P_{14} + P_{25} + P_{48} + K_1$
$= X_{35} + X_{40} + X_{46} + X_{57}$

$X_{35} + X_{40} + X_{46} + X_{57}$
$= Y_3 + Y_8 + Y_{14} + Y_{25}$

$Y_3 + Y_8 + Y_{14} + Y_{25} + K_2$
$= C_3 + C_8 + C_{14} + C_{25} + C_{48}$

$P_3 + P_8 + P_{14} + P_{25} + P_{48} + K_1 + K_2 = C_3 + C_8 + C_{14} + C_{25} + C_{48}$
With probability?

Piling-up Lemma

- Let X, Y, Z be independent stochastic binary variables
  - $\Pr(X = Y) = p_1$
    - $\Pr(X = 1 + Y) = 1 - p_1$
  - $\Pr(Y = Z) = p_2$
- Then $\Pr(X = Z) = p_1p_2 + (1 - p_1)(1 - p_2)$

Proof:

- $\Pr(X = Y \text{ and } Y = Z) = p_1p_2$
- $\Pr(X = Y \text{ and } Y = 1+Z) = p_1(1 - p_2)$
- $\Pr(X = 1+Y \text{ and } Y = Z) = (1 - p_1)\, p_2$
- $\Pr(X = 1+Y \text{ and } Y = 1+Z) = (1 - p_1)(1 - p_2)$

With biases

- $\Pr(X = Z) = p_1p_2 + (1 - p_1)(1 - p_2)$
- Replace p_1 by $\varepsilon_1 + ½$, p_2 by $\varepsilon_2 + ½$
- $\Pr(X = Z) = ½ + 2\varepsilon_1\varepsilon_2$
- For t independent variables:

$$\varepsilon_{1\ldots t} = 2^{t-1} \prod_i \varepsilon_i$$

Correlation

- Correlation between two Boolean functions f(x), g(x)

$$C(f,g) = \Pr(f(x) = g(x)) - \Pr(f(x) \neq g(x))$$
$$= 2\Pr(f(x) = g(x)) - 1$$

- Correlation = 2 $\times$ bias
- Correlation theorem:

$$C(X_1,X_3) = C(X_1,X_2) \times C(X_2,X_3)$$

- More elegant and more standard (outside cryptography)

Using approximations: Algorithm 1

- We have a relation
  P[i] + C[j] = K[k] with prob. p
- With
  - P[i]: sum of some plaintext bits
  - C[j]: sum of some ciphertext bits
  - K[k]: sum of some key bits
- Collect N plaintext-ciphertext pairs
  - Count how many times K[k] = 0,1 is suggested
- For $N > (p - ½)^{-2}$, with high probability the correct value equals the most often suggested value

Algorithm 2

- Guess round key bits of last round
- Apply Algorithm 1 on first r-1 rounds
  - Relation between plaintext, key and guess-decrypted ciphertext
- Observation: Algorithm 1 will be successful only if the guessed key bits are correct
  - We can obtain many more bits
  - Need to approximate one round less
- Extension: guess also first round key

Finding good approximations

- We need approximations with high bias
- Surprisingly similar to the problem of finding good characteristics in differential cryptanalysis
- *Linear characteristics*
- *Linear probability* = $(2p - 1)^2$

Further concepts of linear cryptanalysis

- Differential $\rightarrow$ Linear hull
- $ELP(a,b) = \sum_Q ELP(Q) = \sum_Q \prod_i ELP(\text{round } i)$
- For “iterative cipher with key addition”:
  LP[k](Q) = ELP(Q)
  - Hypothesis of S.E. holds for linear characteristics
  - (Assuming independent round keys)
- LP[k](a,b) remains a problem

Comparison

Linear cryptanalysis	**Differential cryptanalysis**
Known plaintexts	Chosen plaintexts
Statistics on large set of texts	1 Pair of texts (repeated)
Needs more texts	Less texts (exception: DES)

Attacks are remarkably similar in most points

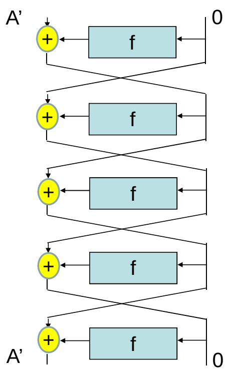
Variations on Differential Cryptanalysis

Vincent Rijmen

Overview

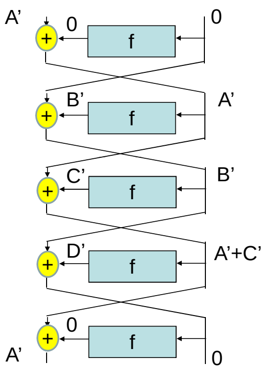
- Impossible differentials
  - Attack on Feistel ciphers
- Saturation attack

A 5-round differential



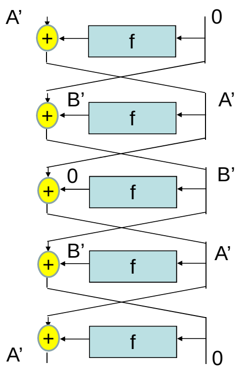
Compute the EDP of the differential (A',0)→(A',0) when f is an injective function

The 5-round characteristics



Conditions:
D' + B'= 0
A' + C' = A'

The 5-round characteristics



Conditions:
D' = B'
C' = 0

Probability

$EDP_{cipher}(A'0 \rightarrow A'0)$
$= \sum_{B'} (EDP_f(A' \rightarrow B'))^2 \times EDP_f(B' \rightarrow 0)$

If f is injective function,

Then $EDP_f (B' \rightarrow 0) > 0 \Leftrightarrow B' = 0$
But if $B' = 0$
Then $EDP_f (A' \rightarrow B') > 0 \Leftrightarrow A' = 0$

Hence, if $A' \neq 0$, then the 5-round differential has probability 0

Impossible differentials

- Differential with probability 0 means that there are *no* right pairs
  - If we think we see a right pair, then we made a wrong assumption
- Attack on 6 rounds:
  - Encrypt pairs of plaintext with difference (A',0)
  - Guess last round key and decrypt ciphertexts one round
  - If we find difference (A',0) at the guessed output of round 5, then this guess for the key must be wrong

Data complexity

- One attempt will detect a wrong key with probability 2^{-n}
- Encrypt the $2^{n/2}$ texts with right half constant
  - $2^{n/2}(2^{n/2}-1)/2 \approx 2^{n-1}$ pairs
  - This will suffice to detect 50% of the wrong keys
  - Repeat z times to eliminate all wrong keys
    - z = round key length

Computational complexity

- We need to try out each round key at least once:
  - (Academic) attack only if round key is shorter than master key
  - Typically OK with Feistel ciphers
- Observation:
  - (A',0) after 5th round can only happen if (X',A') after 6th round
  - Only for these pairs we need to try out the keys

Saturation attack

- First `Square attack' [Daemen, Rijmen & Knudsen '98]
- Later: SASAS, integral cryptanalysis, saturation
- Chosen plaintext attack
  - Texts chosen in larger groups

Saturation attack basics

- Focus on AES
- Λ-set: set of 256 states a_t (4x4 byte arrays) such that for all indices i,j:
  - $a_t[i,j] = c \; \forall \; t$ (*passive*, *constant*), or
  - $a_t[i,j] \neq a_s[i,j]$ if $t \neq s$ (*active*, *saturated*), or
  - $\sum_t a_t[i,j] = 0$ (*balanced*)
- (active implies balanced, constant implies balanced)

Example 1

- Consider Λ-set where $a_t[0,0] = t$, and for other i,j: $a_t[i,j] = 0$.
- What do we know about $b_t = AK_k(a_t)$?
- About $c_t = SB(b_t)$?
- About $d_t = SR(c_t)$?
- About $e_t = MC(d_t)$?

Example 2

- Λ-set where $a_t[0,0] = a_t[1,1] = t$, and for other i,j: $a_t[i,j] = 0$.
- What do we know about $b_t = AK_k(a_t)$?
- About $c_t = SB(b_t)$?
- About $d_t = SR(c_t)$?
- About $e_t = MC(d_t)$?

Action of AES step transformations on a Λ-set

- ShiftRows: only changes the indices [i,j]
- SubBytes:
  - Saturated bytes remain saturated
  - Constant bytes remain constant
  - Balanced bytes become undetermined
- AddRoundKey:
  - Saturated bytes remain saturated
  - Constant bytes remain constant
  - Balanced bytes remain balanced

Action of MixColumns on a Λ-set

- Action depends on all 4 bytes of the column:

$$b_t[i,j] = c[i,0]a_t[0,j] + c[i,1]a_t[1,j] + \cdots$$

  - [CCCC] $\rightarrow$ [CCCC]
  - [CCCS] $\rightarrow$ [SSSS]

$$\begin{aligned} \sum_t b_t[i,j] &= \sum_t (c[i,0]a_t[0,j] + c[i,1]a_t[1,j] + \cdots) \\ &= c[i,0]\sum_t a_t[0,j] + c[i,1]\sum_t a_t[1,j] + \cdots \end{aligned}$$

  - [BBBB] $\rightarrow$ [BBBB]

A 3-round distinguisher for AES

$$\begin{bmatrix} S & C & C & C \\ C & C & C & C \\ C & C & C & C \\ C & C & C & C \end{bmatrix} \xrightarrow{\text{SB}} \begin{bmatrix} S & C & C & C \\ C & C & C & C \\ C & C & C & C \\ C & C & C & C \end{bmatrix} \xrightarrow{\text{SR}} \begin{bmatrix} S & C & C & C \\ C & C & C & C \\ C & C & C & C \\ C & C & C & C \end{bmatrix} \xrightarrow{\text{MC}} \begin{bmatrix} S & C & C & C \\ S & C & C & C \\ S & C & C & C \\ S & C & C & C \end{bmatrix} \xrightarrow{\text{AK}}$$

$$\begin{bmatrix} S & C & C & C \\ S & C & C & C \\ S & C & C & C \\ S & C & C & C \end{bmatrix} \xrightarrow{\text{SB}} \begin{bmatrix} S & C & C & C \\ S & C & C & C \\ S & C & C & C \\ S & C & C & C \end{bmatrix} \xrightarrow{\text{SR}} \begin{bmatrix} S & C & C & C \\ C & C & C & S \\ C & C & S & C \\ C & S & C & C \end{bmatrix} \xrightarrow{\text{MC}} \begin{bmatrix} S & S & S & S \\ S & S & S & S \\ S & S & S & S \\ S & S & S & S \end{bmatrix} \xrightarrow{\text{AK}}$$

$$\begin{bmatrix} S & S & S & S \\ S & S & S & S \\ S & S & S & S \\ S & S & S & S \end{bmatrix} \xrightarrow{\text{SB}} \begin{bmatrix} S & S & S & S \\ S & S & S & S \\ S & S & S & S \\ S & S & S & S \end{bmatrix} \xrightarrow{\text{SR}} \begin{bmatrix} S & S & S & S \\ S & S & S & S \\ S & S & S & S \\ S & S & S & S \end{bmatrix} \xrightarrow{\text{MC}} \begin{bmatrix} B & B & B & B \\ B & B & B & B \\ B & B & B & B \\ B & B & B & B \end{bmatrix} \xrightarrow{\text{AK}}$$

Attacking 4 rounds

- AES-4 = AK + 3 rounds + final round
- Initial AK doesn't matter
- Final round: no MixColumns
- Attack:
  - Encrypt a Λ-set with one saturated position
  - Guess 1 byte of last round key
  - Decrypt 1 byte of output of 3rd round
  - Verify Balance property
    - For correct guess of the key, property must hold
    - For incorrect guess, property holds with prob. 1/256
- Presence of MixColumns in last round wouldn't help

Adding a round at the end

- AES-5 = AK + 3 rounds + 1 round + final round
- Guess 1 column of key in 5th round
- Decrypt one byte of output of 4th round
- Apply previous attack

- We need approx. 6 Λ-sets and 2^{40} steps

Conclusions

- Block cipher design is still a craft rather than a science
- Plenty of interesting open questions

- Would benefit from more attention by mathematicians with one eye open for practice