Analysis of XSL Applied to BES

By: Lim Chu Wee,
Khoo Khoong Ming.

History

- (2002) Courtois and Pieprzyk announced a plausible attack (XSL) on Rijndael AES.
  - Complexity of $\approx 2^{225}$ for AES-256.
- Later Murphy and Robshaw proposed embedding AES into BES, with equations over F_{256}.
  - S-boxes involved fewer monomials, and would provide a speedup for XSL *if it worked* (2^{87} for AES-128 in best case).
  - Murphy and Robshaw also believed XSL *would not work*.
- (Asiacrypt 2005) Cid and Leurent showed that "compact XSL" does not crack AES.

Summary of Our Results

- We analysed the application of XSL on BES.
- Concluded: the estimate of 2^{87} was too optimistic; we obtained a complexity $\geq 2^{401}$, *even if XSL works*. Hence it does not crack BES-128.
- Found further linear dependencies in the expanded equations, upon applying XSL to BES.
  - Similar dependencies exist for AES – unaccounted for in computations of Courtois and Pieprzyk.
- Open question: does XSL work at all, for some P?

Quick Description of AES & BES

AES Structure

- Very general description of AES (in F_{256}):
  - Input: key ($k_0k_1\ldots k_{s-1}$), message ($M_0M_1\ldots M_{15}$).
  - Suppose we have aux variables: v_0, v_1, ….
  - At each step we can do one of three things:
    - Let v_i be an F_2-linear map T of some previously defined byte: one of the v_j's, k_j's or M_j's.
    - Let v_i = XOR of two bytes.
    - Let v_i = S(some byte).
  - Here S is given by the map: $x \to x^{-1}$ (S(0)=0).
  - Output = 16 consecutive bytes $v_{i-15}\ldots v_{i-1}v_i$.

BES Structure

BES writes all equations over F_{256}.

- For each $v \in F_{256}$, we also include its conjugates:
  - i.e. $v, v^2, v^4, v^8, v^{16}, v^{32}, v^{64}, v^{128}$ ($v^{256} = v$).
- Then an F_2-linear map $y = T(v)$ can be written as an F_{256}-linear map of $v, v^2, \ldots v^{128}$.
  - Conjugates of y can also be written in this manner.
- S-box has a simple expression: $v_i = v_j^{-1}$.
  - For conjugate, $v_i^2 = (v_j^2)^{-1}$.
- For XOR, conjugates give $(v_i+v_j)^2 = (v_i^2)+(v_j^2)$.

Summary of XSL on AES / BES (and Notations)

XSL on AES

- Write all equations over F_2.
- *Including key schedule,*
  - AES-128 has **S=201** S-boxes, **L=1664** linear eqns;
  - AES-192 has **S=417** S-boxes, **L=3520** linear eqns;
  - AES-256 has **S=501** S-boxes, **L=4128** linear eqns.
- If $(y_0 y_1 \ldots y_7) = S(x_0 x_1 \ldots x_7)$, then the x_i's and y_i's satisfy r=24 "bilinear" equations,
  - involving t=81 monomials: 1, x_i, y_j, $x_i y_j$.
- Let P = XSL parameter.

- Form the set Σ_S of **extended S-box** equations as follows:
  - Pick 1 *active* S-box, P-1 *passive* S-boxes (all S-boxes distinct).
  - Pick an equation from active S-box, one S-box monomial from each passive S-box.
  - Multiply the equation by these P-1 monomials.
- Form the set Σ_L of **extended linear** equations as follows:
  - Pick 1 linear equation, P-1 distinct *passive* S-boxes.
  - Pick a monomial from each passive S-box.
  - Multiply the equation by these P-1 monomials.
- Collect these equations $\Sigma_S \cup \Sigma_L$.
- Solve the equations via linearisation: replace each monomial with new variable and solve linearly.

- Courtois & Pieprzyk noted some obvious linear dependencies:
  - Pick 2 active S-boxes, and S-box equations eqn_1 and eqn_2.
  - Pick P-2 passive S-boxes, and S-box monomials $t_3, \ldots t_P$.
  - Expanding $(eqn_1)(eqn_2)(t_3 \ldots t_P)$, we get a linear relation between equations extended from eqn_1 and those from eqn_2.
- Eliminating these linear dependencies,
  - number of extended S-box equations $R = C(S, P)\ (t^P-(t-r)^P)$,
  - number of extended linear eqns $R' = L\ (t-r)^{P-1}\ C(S, P-1)$.
- *Note: we have combined R' and R" in Courtois' & Pieprzyk's paper into a single R' here.*

- On the other hand, number of monomials $T = t^P C(S,P)$.
- We want more equations than monomials. Hence,
  - **AES-128** : min P = 7. This gives $R = 4.95 * 10^{25}$, $R' = 4.85 * 10^{24}$ and $T = 5.41 * 10^{25}$. Complexity of XSL = $T^{2.376} = 2^{203}$.
  - **AES-192** : min P = 7. This gives $R = 8.65 * 10^{27}$, $R' = 8.50 * 10^{26}$ and $T = 9.46 * 10^{27}$. Complexity of XSL = $T^{2.376} = 2^{221}$.
  - **AES-256** : min P = 7. This gives $R = 3.15 * 10^{28}$, $R' = 3.02 * 10^{27}$ and $T = 3.45 * 10^{28}$. Complexity of XSL = $T^{2.376} = 2^{225} < 2^{256}$.
- "T'-method": multiply equations by monomials selectively, without increasing its degree – to get more equations.
  - To apply T', need at least 0.994 of needed equations.
- It seemed plausible that XSL can break AES-256 faster than brute force.

XSL on BES

- For each variable v, write v_0, v_1, … v_7 for the conjugates of v.
- Hence, for each S-box y = S(x), we get r=24 equations:
  - $x_i y_i = 1$, i=0,1,…,7;
  - $y_i^2 = y_{i+1}$, i=0,1,…,7 ($y_8 = y_0$);
  - $x_i^2 = x_{i+1}$, i=0,1,…,7 ($x_8 = x_0$).
- Monomials appearing: 1, x_i, y_i, $x_i y_i$, x_i^2, y_i^2 (t=41).
- If we apply XSL to BES, then all computations hold, *with t=81 replaced with t=41*. Result: we can use a smaller P.
- E.g. **BES-128**: P=3. This gives $R=8.53 * 10^{10}$, $R' = 9.67 * 10^9$ and $T = 9.19 * 10^{10}$. Complexity = $T^{2.376} = 2^{87} < 2^{128}$ (!!).
- Finally, T'-method cannot be applied to BES.

Our Analysis of XSL on BES

Analysing Extended S-box Eqns (I)

- In BES, all S-box equations are equalities between:

$$x_i y_i = 1,\ \ x_i^2 = x_{i+1},\ \ y_i^2 = y_{i+1}.$$

- Thus, an extended S-box equation is also an equality between two monomials.
- Hence solving them linearly gives equivalence classes of monomials. E.g.
  - suppose $(b_i) = S(a_i)$, $(d_i) = S(c_i)$, $(f_i) = S(e_i)$;
  - $a_2^2 d_4 e_5 f_5 = a_3 d_4 e_5 f_5 = a_3 d_4$, where first equality extended from $a_2^2 = a_3$, second equality from $e_5 f_5 = 1$.
- *In each equivalence class, there is a unique monomial of the form $v^{(1)}v^{(2)}\ldots v^{(i)}$, where the $v^{(j)}$ are variables belonging to different S-boxes.* We will call such S-box monomials **reduced**.

Analysing Extended S-box Eqns (II)

- Number of reduced monomials of degree *i* is: $C(S,i)\ 16^i$.
- Hence, after solving the extended S-box equations by linearisation, we get exactly:

$$\sum_{i=0}^{P} C(S,i) 16^i$$

linearly independent monomials.
- Prior XSL estimate: after eliminating obvious linear dependencies, we get

$$T - R = (t - r)^P C(S,P) = 17^P C(S,P)$$

linearly independent monomials, which is a slight overestimate but rather close.

Analysing Extended Linear Eqns

- Extended linear eqns are obtained by multiplying linear equation with S-box monomials.
- By previous 2 slides, suffices to multiply the linear equation by *reduced* S-box monomials.
- Hence, XSL is equivalent to the following:
  - (a) Pick set Σ_S of extended S-box equations.
  - (b) Pick set Σ_L' of equations which are extended from linear equations by a reduced monomial of degree at most P-1.
  - (c) Solve $\Sigma_S \cup \Sigma_L$' via linearisation.
- *Question: what if we skip the step (a), i.e. forget all extended S-box equations? How many linearly independent monomials do we get?*

Answer (lower bound) to previous slide's question:

- We end up multiplying linear equations by reduced monomials and solving by linearisation.
- Recall the original description of AES, where each byte is defined in terms of previous defined bytes. *Key point: upon removal of the S-boxes, we introduce 8S (totally) free F_{256} variables (i.e. these 8 variables can take any value).*
- Nutshell: by skipping step (a), we introduce 8S totally free variables – which we can take to be the input variables.
- The number of linearly independent monomials is hence *at least* number of reduced monomials formed by these 8S variables:

$$D_1 = \sum_{i=0}^{P} C(S,i)8^i$$

- Big question : *does adding step (a) provide enough equations to remove these linear independence?*
- Recall: adding step (a) serves to replace every S-box monomial by a reduced monomial.
- Since an equation in Σ_L' is of the form (eqn)*(reduced monomial), the only useful extended S-box equations are of the form:

$$(v)(\text{monomial}_1) = (\text{monomial}_2),$$

  - where monomial_1 is a reduced monomial of deg $\leq$ P-1,
  - v is a variable occuring in monomial_1, or whose dual occurs in monomial_1,
  - monomial_2 is a reduced monomial,
  - *furthermore, we can assume other than the dual/identical pair, all other variables in monomial_1 are input variables,*
  - if $(b_i) = S(a_i)$, $(d_i) = S(c_i)$, $(f_i) = S(e_i)$, then an example would be $(e_2)(a_2c_7f_2) = (a_2c_7)$.

- Let us count the number of such useful S-box equations:

$$D_2 = 24S \times \sum_{i=0}^{P-2} C(S-1, i)8^i$$

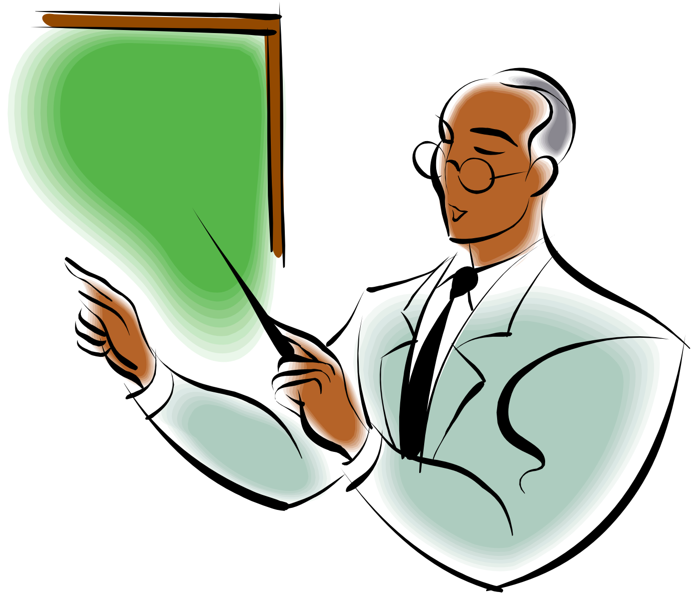
- For linearisation to work, we must have $D_2 \geq D_1$.
- We get the following values:
  - **BES-128** : min P = 23. $D_1 = 5.90 * 10^{50}$, $D_2 = 6.25 * 10^{50}$. Resulting complexity = $D_1^{2.376} = 2^{401}$.
  - **BES-192** : min P = 33. $D_1 = 5.86 * 10^{78}$, $D_2 = 6.02 * 10^{78}$. Resulting complexity = $D_1^{2.376} = 2^{622}$.
  - **BES-256** : min P = 36. $D_1 = 3.80 * 10^{78}$, $D_2 = 3.85 * 10^{78}$. Resulting complexity = $D_1^{2.376} = 2^{691}$.

- **Conclusion, XSL does not break BES faster than brute force.**

Further Analysis

- Our analysis shows a lot of linear dependencies previously unaccounted for.
- *Observation 1 : Original computations assumed that only extended S-box monomials appear.*
  - Not true. E.g. suppose $y = S(x)$ is an S-box. A linear equation contains x_2, then this S-box appears as a passive one, with y_5 chosen, then the monomial contains a factor of x_2y_5 – which is not from S-box.
  - Heuristically, difference not significant.

- *Observation 2 : “Obvious” linear dependencies among extended linear equations*.
  - E.g. if L_1 and L_2 are linear equations, and $v_3,\dots v_P$ are monomials from P-2 distinct S-boxes.
  - Expanding $L_1L_2(v_3\dots v_p)$ forms a linear dependence between equations extended from L_1 and those from L_2.
  - Similar to linear dependencies among extended S-box equations, but were not accounted for.
  - Likely to be very significant, as demonstrated by those among extended S-box equations.
- **Based on these observations, we believe that XSL is unlikely to work on AES over F_2, or on Serpent.**



Thank you.

Questions?