CS641

Modern Cryptology
Indian Institute of Technology, Kanpur

Group Name: Cipherberg
Deeksha Arora (20111017), Sambhrant Maurya (20111054), Shruti Sharma (20111061)

Mid Semester Examination

Date of Submission:
March 10, 2021

Question 1

Consider a variant of DES algorithm in which the S-box S1 is changed as follows:

> For every six bit input α, the following property holds: $\mathrm{S1}(\alpha) = \mathrm{S1}(\alpha \oplus 001100) \oplus 1111$.

All other S-boxes and operations remain the same. Design an algorithm to break four rounds of this variant. In order to get any credit, your algorithm must make use of the changed behavior of S1.

Solution

In the question it is mentioned that for every 6 bit input α, S1 box holds the following property:

$$\mathrm{S1}(\alpha) = \mathrm{S1}(\alpha \oplus 001100) \oplus 1111 \tag{1.1}$$

Therefore, if α and α' are two 6 bit inputs to the $S1$ box such that $\alpha \oplus \alpha' = 001100$ then,

$$\mathrm{S1}(\alpha) \oplus \ \mathrm{S1}(\alpha') = 1111 \tag{1.2}$$

Cryptanalysis of 4 round DES:
We use differential cryptanalysis to recover the key. If we select random input pairs that have input XOR to second round S1 as 001100, then we expect that with probability 1, the XOR of the output will be 1111.

Step 1: Generate plaintext pairs $\mathrm{L_0R_0}$ and $\mathrm{L'_0R'_0}$ such that $\mathrm{R_0} = \mathrm{R'_0}$ and input XOR to second round S1 box is 001100 and input XOR to remaining S-boxes in second round is all zeroes. This allows us to predict the XOR of the second round output of S-boxes with probability 1. (Reason: From properties of S1 box, it can be inferred that XOR of round 2 output of S1 box for this plaintext pair will be 1111)

Step 2: For the plaintext pair $(\mathrm{L_0R_0}, \mathrm{L'_0R'_0})$, obtain the corresponding ciphertext pair $(\mathrm{L_4R_4}, \mathrm{L'_4R'_4})$ by getting the sender to encrypt this plaintext pair (chosen plaintext attack).

Step 3: Let $E(R_3) = \beta_1\beta_2\cdots\beta_8$ and $E(R'_3) = \beta'_1\beta'_2\cdots\beta'_8$ with $|\beta_i| = 6 = |\beta'_i|$, where $\mathrm{R_3}$ and $\mathrm{R'_3}$ are right-halves of output of third round on the plaintexts L_0R_0 and $L'_0R'_0 = L'_0R_0$.

Let $\alpha_i = \beta_i \oplus k_{4,i}$ and $\alpha'_i = \beta'_i \oplus k_{4,i}$ where $|\alpha_i| = 6 = |\alpha'_i|$ and $k_4 = k_{4,1}k_{4,2}\cdots k_{4,8}$.

Let $\gamma_i = S_i(\alpha_i)$ and $\gamma'_i = S_i(\alpha'_i)$, where $|\gamma_i| = 4 = |\gamma'_i|$.

Here, we know β_i, β'_i and $\beta_i \oplus \beta'_i = \alpha_i \oplus \alpha'_i$. Also, the value $\gamma = \gamma_i \oplus \gamma'_i$ is known with probability 1.

Now, to find key $k_{4,i}$ we need to find the sets X_i and K_i such that,

$$X_i = \left\{ (\alpha, \alpha') \mid \alpha \oplus \alpha' = \alpha_i \oplus \alpha'_i \text{ and } S_i(\alpha) \oplus S_i(\alpha') = \gamma \right\} \tag{1.3}$$

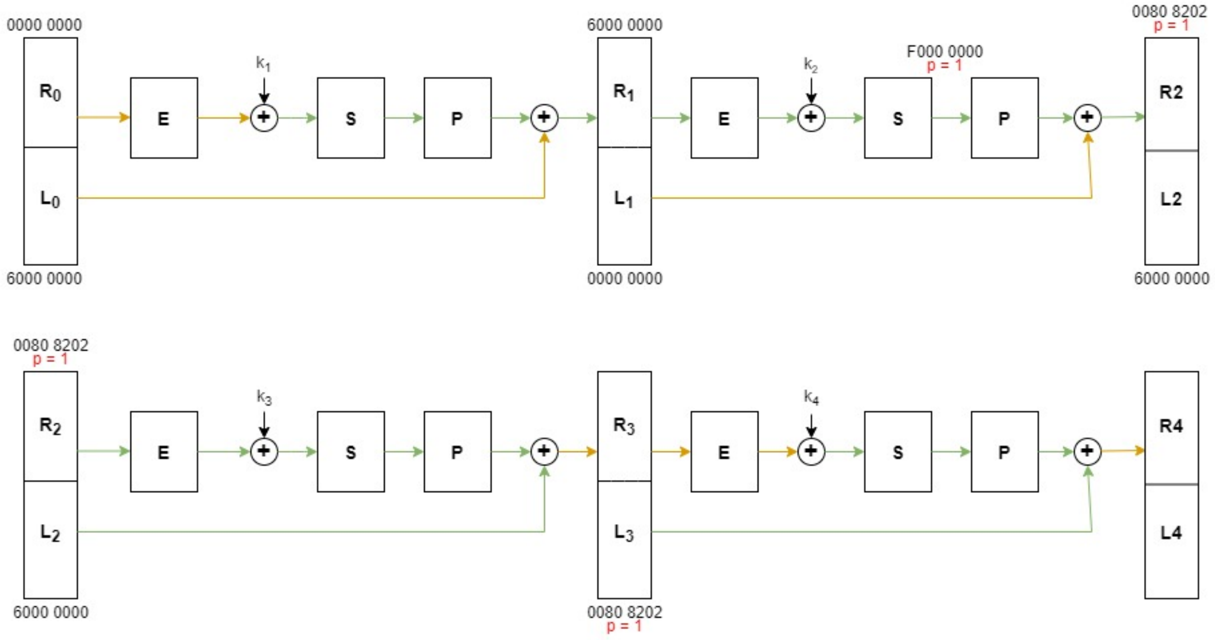
$$K_i = \left\{ k \mid \beta_i \oplus k = \alpha \text{ and } (\alpha, \alpha') \in X_i \text{ for some } \alpha' \right\} \tag{1.4}$$

From equation 1.3, we can say that the pair $(\alpha_i, \alpha'_i) \in X_i$ and therefore, $k_{4,i} \in K_i$.
(Reference: slide 8, Lecture 6)

Step 4: Repeat steps 1, 2 and 3 for multiple pairs of plaintexts until the key $k_{4,i}$ is uniquely identified.
Note: The key $k_{4,i}$ can be uniquely identified by finding the intersection of the sets $K_{i,1}K_{i,2}\cdots K_{i,l}$ obtained for l different plaintext pairs since the correct key value $k_{4,i}$ must occur (due to probability being 1) in each set $K_{i,l}$.

By carrying out the above steps, we can uniquely identify the key k_4.

The following image shows the how the XOR of a pair of input text block travels through 4 round DES.



In above figure, texts in all lines marked orange are known and for lines marked green, XOR of input text blocks is known.

Step 6: : At this point we know 48 bits of the key which comprises $k_{4,1} \cdots k_{4,8}$. The remaining 8 bits of the key can be found by brute force.

In the above algorithm, the 2 round characteristic used to break 4 round DES is $(6000\overline{0}, \overline{0}\,\overline{0}, 1, \overline{0}\,\overline{0}, 6000\overline{0}, 1, 6000\overline{0}, 00808202)$

Question 2

The SUBSET-SUM problem is defined as follows:

> Given $(a_1, \ldots, a_n) \in \mathbb{Z}^n$ and $m \in \mathbb{Z}$, find $(b_1, \ldots, b_n) \in \{0,1\}^n$ such that $\sum_{i=1}^{n} a_i b_i = m$ if it exists.

This problem is believed to be a hard-to-solve problem in general. Consider a hypothetical scenario where Anubha and Braj have access to a fast method of solving SUBSET-SUM problem. They use the following method to exchange a secret key of AES:

> Anubha generates an $n = 128$ bit secret key k. She then chooses n positive integers $a_1, \ldots, a_n$ such that $a_i > \sum_{1 \le j < i} a_j$. She computes $m = \sum_{i=1}^{n} a_i k_i$ and sends $(a_1, a_2, \ldots, a_n, m)$ to Braj, where k_i is ith bit of k. Upon receiving numbers $(a_1, a_2, \ldots, a_n, m)$, Braj solves the SUBSET-SUM problem to extract the key k.

Show that an attacker Ela does not need to solve SUBSET-SUM problem to retrieve the key k from $(a_1, a_2, \ldots, a_n, m)$.

Solution

$(a_1, \ldots, a_n) \in \mathbb{Z}^n$ such that $a_i > \sum_{1 \le j < i} a_j$ is a superincreasing sequence. We propose a $O(n)$ linear time greedy algorithm for any $n \in \mathbb{N}$, which Ela can use as an attacker to calculate the 128 bit secret key k from the information $(a_1, a_2, \ldots, a_n, m)$ sniffed from the channel, where $m = \sum_{i=1}^{n} a_i k_i$. This problem is quite analogous to the Merkle-Hellman knapsack cryptosystem.

1. Let $A = (a_1, \ldots, a_n)$, $m' = m$ and X be a list of size n initialized to all zeroes, i.e. $X = [0, 0, 0, \text{...n times}]$.

2. For i= n to 1, do:

    2.1 If a_i is less than or equal to m' then, set $X[i] = 1$ and $m' = m' - a_i$.

    2.2 If $m' == 0$, break.

The list X gives the key k.

Let's take a small example to demonstrate the algorithm with $n = 5$.

Let $A = (a_1, \ldots, a_n) = (1, 2, 4, 8, 16)$ and let $k = 01101$. m found on the channel will be $m = \sum_{i=1}^{n} a_i k_i = 2 * 1 + 4 * 1 + 16 * 1 = 22$. Initialize $X = [0, 0, 0, 0, 0]$ and $m' = m = 22$.

Iteration 1: $m' = 22$ and $i = 5$. $a_5 = 16 \leq m'$. Therefore, set $X[5] = 1$. Hence $X = [0, 0, 0, 0, 1]$. Set $m' = 22 - 16 = 6$.

Iteration 2: $m' = 6$ and $i = 4$. $a_4 = 8 > m'$. Therefore, $X[4] = 0$ and m' will not change.

Iteration 3: $m' = 6$ and $i = 3$. $a_3 = 4 \leq m'$. Therefore, set $X[3] = 1$. Hence $X = [0, 0, 1, 0, 1]$. Set $m' = 6 - 14 = 2$.

Iteration 4: $m' = 2$ and $i = 2$. $a_2 = 2 \leq m'$. Therefore, set $X[2] = 1$. Hence $X = [0, 1, 1, 0, 1]$. Set $m' = 2 - 2 = 0$.

Since $m = 0$, the iterations stop.

The list $X = [0, 1, 1, 0, 1]$ gives the key k, i.e. $k = 01101$.

Proof for Correctness of Algorithm:
Statement: On traversing the sequence $(a_1, \ldots, a_n) \in \mathbb{Z}^n$ where $a_i > \sum_{1 \leq j < i} a_j$, from index i=n to 1, if an element a_i is less than or equal to m′ then $k_i = 1$, where m' is defined in the algorithm above.

Proof by Contradiction
Let's assume an element $a_i \in (a_1, \ldots, a_n)$ such that $a_i <= m'$ and $a_{i+1} > m'$ and $k_i \neq 1$. We know that, $a_i > \sum_{1 \leq j < i} a_j$, therefore even if all the elements from j=1 to (i-1) are added then also their sum will never be equal to m'. In other words,
$\sum_{j=1}^{i-1} a_j k_j < m' \implies \sum_{j=1}^{n} a_j k_j < m$.
But this contradicts the fact that $m = \sum_{i=1}^{n} a_i k_i$. Therefore, our assumption is wrong. Hence, if an element a_i is such that $a_i \leq m'$ and $a_{i+1} > m'$ then $k_i = 1$.

This shows that Ela can easily calculate the AES key k using the above $O(n)$ algorithm without solving the subset sum problem.

Reference: Merkle-Hellman knapsack cryptosystem:
https://en.wikipedia.org/wiki/Merkle%E2%80%93Hellman_knapsack_cryptosystem

Question 3

Having failed to arrive at a secret key as above, Anubha and Braj try another method. Let G be the group of $n \times n$ invertible matrices over field F, $n = 128$. Let $a, b, g \in G$ such that $ab \neq ba$. The group G and the elements a, b, g are publicly known. Anubha and Braj wish to create a shared secret key as follows:

> Anubha chooses integers ℓ, m randomly with $1 < \ell, m \leq 2^n$, and sends $u = a^\ell g b^m$ to Braj. Braj chooses integers r, s randomly with $1 < r, s \leq 2^n$, and sends $v = a^r g b^s$ to Anubha. Anubha computes $k_a = a^\ell v b^m = a^{\ell+r} g b^{m+s}$. Braj computes $k_b = a^r u b^s = a^{\ell+r} g b^{m+s}$. The secret key is thus $k = k_a = k_b$.

Show that even this attempt fails as Ela can find k using u and v.

Hint: Show that Ela can

1. find elements x and y such that $xa = ax$, $yb = by$, and $u = xgy$,
2. use x, y, and v to compute k.

Solution

G is a finite group of n*n invertible matrices over a finite field F, n=128. Public elements are $a, b, g \in G$ such that $ab \neq ba$. The scenario in the question compares with the standard Stickel Key Agreement Protocol except that Stickel, in his paper, took n=31.

Anubha chooses l, m uniformly at random, such that, $1 < l, m \leq 2^n$ and sends $u = a^l g b^m$ to Braj.
Braj chooses r, s uniformly at random, such that, $1 < r, s \leq 2^n$ and sends $v = a^r g b^s$ to Anubha.
Anubha calculates $k_a = a^l v b^m = a^{l+r} g b^{m+s}$
Braj calculates $k_b = a^r u b^s = a^{r+l} g b^{s+m}$
So, Anubha and Braj have same group element and the shared secret key is $k = k_a = k_b$

This approach is vulnerable to linear algebra attacks. Ela, the trespasser, needs to solve a decomposition search problem in which she has to solve an equation of the form $\alpha = x\beta$y and find $x, y \in G$ when α and β are known[1].

Ela doesn't need to recover any of the private exponents l, m, r, s to derive key k. Instead it is sufficient to intercept transmitted messages u and v to find n*n matrices $x, y \in G$ such that,

$$xa = ax, yb = by, and\ u = xgy \tag{3.1}$$

Ela can then find the key k using following matrix algebra *[proof at end]* ,

$$xvy = xa^r gb^s y = a^r xgyb^s = a^r ub^s = k_b = k$$

The system of equations in (3.1) is a nonlinear system which translates to $3n^2$ equations in $2n^2$ unknowns. We need to solve these equations to find the unknowns x and y.

The first two equations in (3.1), $xa = ax$ and $yb = by$, translate to a system of n^2 linear equations for unknown entries of matrices x and y where a and b are known.
Equation $u = xgy$ is not linear due to the presence of product of two unknown matrices x and y, but since we know that $x \in G$ thus, it is invertible, so we can re-arrange by multiplying both sides of $u = xgy$ by x^{-1} on left i.e. $x^{-1}u = gy$ where g and u are known.

Since $xa = ax$ iff $x^{-1}a = ax^{-1}$ *[proof at end]* so we denote $x_1 = x^{-1}$ and solve following system of equations.

$$x_1 a = ax_1, yb = by, and\ x_1 u = gy \tag{3.2}$$

The system of equations in (3.2) is a linear system where we need to solve for x_1 and y. Each equation translates to a system of n^2 linear equations for unknown entries of matrices x_1 and y. Therefore, now there are $3n^2$ linear equations with $2n^2$ unknowns. A solution to (3.2) is guaranteed since the equations are always satisfied by $x_1 = (a^l)^{-1}$ and $y = b^m$, and may be found by Gaussian elimination[3].

Since u is a known invertible matrix, we multiple both sides of $x_1 u = gy$ by u^{-1} on right, to get, $x_1 = gyu^{-1}$, and now we can eliminate x_1 to solve equations,

$$gyu^{-1}a = agyu^{-1}, yb = by \tag{3.3}$$

Now, the only unknown is y, and we have $2n^2$ linear equations for n^2 entries of y. We need to solve this heavily overdetermined system of linear equations in (3.3) to find the invertible matrix y (n=128, so $n^2 = 16384$). This system must have atleast one non-trivial/non-zero solution. Reduce the matrix of this system to an echelon form and since it is a heavily overdetermined system, the number of free variables will not be too big. Therefore, we can go over possible values of free variables one at a time until we find values that yield an invertible matrix.

After getting y, x can be found out by solving the equations $x_1 = gyu^{-1}$ and $x_1 = x^{-1}$. The key k can now be computed as $k = xvy$. Thus Ela can find the key without knowing any of the private components.

Some trivial results of Matrix Algebra used in Solution 3

1.

$$\begin{aligned} xa &= ax & & \\ xa^2 &= a(xa) & & \text{multiply by a on right} \\ xa^2 &= a^2x & & \text{since, ax = xa} \\ xa^n &= a^nx & & \text{similarly, for n times} \end{aligned}$$

2.

$$\begin{aligned} if,\ x^{-1}a &= ax^{-1} & & \\ xx^{-1}a &= xax^{-1} & & \text{multiply by x on left} \\ a &= xax^{-1} & & \text{since, } xx^{-1} = I \\ ax &= xax^{-1}x & & \text{multiply by x on right} \\ ax &= xa & & \text{since, } x^{-1}x = I \end{aligned}$$

References

[1] R. Blackburn, C. Cid and C. Mullan, Group theory in cryptography, 2009

[2] V. Shpilrain, Cryptanalysis of Stickel's Key Exchange Scheme, 2008

[3] Ciaran Mullan, Cryptanalysing variants of Stickel's key agreement scheme , 2011