ISE334/SE425: E-Commerce, Com., and Info. Security	Recitation 3
Semester 2 5774	12 March 2014

Simplified DES

1 Introduction

In this lab we will work through a simplified version of the DES algorithm. The algorithm is not cryptographically secure, but its operations are similar enough to the DES operation to give a better feeling for how it works.

We will proceed by reading the Simplified DES algorithm description in the Stallings section. We will then work through a full example in class.

2 Full Example

Let the plaintext be the string 0010 1000. Let the 10 bit key be 1100011110.

2.1 Key Generation

The keys k_1 and k_2 are derived using the functions $P10$, Shift, and $P8$.

$P10$ is defined as follows:

P10									
3	5	2	7	4	10	1	9	8	6

$P8$ is defined to be as follows:

P8							
6	3	7	4	8	5	10	9

The first key k_1 is therefore equal to:

Bit #	1	2	3	4	5	6	7	8	9	10
K	1	1	0	0	0	1	1	1	1	0
$P10(K)$	0	0	1	1	0	0	1	1	1	1
$Shift(P10(K))$	0	1	1	0	0	1	1	1	1	0
$P8(Shift(P10(K)))$	1	1	1	0	1	0	0	1		

The second key k_2 is derived in a similar manner:

Bit #	1	2	3	4	5	6	7	8	9	10
K	1	1	0	0	0	1	1	1	1	0
$P10(K)$	0	0	1	1	0	0	1	1	1	1
$Shift^3(P10(K)))$	1	0	0	0	1	1	1	0	1	1
$P8(Shift^2(P10(K)))$	1	0	1	0	0	1	1	1		

So we have the two keys $k_1 = \{1110\ 1001\}$ and $k_2 = \{1010\ 0111\}$

2.2 Initial and Final Permutation

The plaintext undergoes an initial permutation when it enters the encryption function, IP. It undergoes a reverse final permutation at the end IP^{-1}.

The function IP is defined as follows:

IP							
2	6	3	1	4	8	5	7

The function IP^{-1} is defined as follows:

IP^{-1}							
4	1	3	5	7	2	8	6

Applied to the input, we have the following after the initial permutation:

Bit #	1	2	3	4	5	6	7	8
P	0	0	1	0	1	0	0	0
$IP(P)$	0	0	1	0	0	0	1	0

2.3 Functions f_K, SW, K

- The function f_k is defined as follows. Let $P = (L, R)$, then $f_K(L, R) = (L \oplus F(R, SK), R)$.
- The function SW just switches the two halves of the plaintext, so $SW(L, R) \to (R, L)$
- The function $F(p, k)$ takes a four bit string p and eight bit key k and produces a four bit output. It performs the following steps.

  1. First it runs an expansion permutation E/P:

E/P							
4	1	2	3	2	3	4	1

  2. Then it XORs the key with the result of the E/P function
  3. Then it substitutes the two halves based on the S-Boxes.
  4. Finally, the output from the S-Boxes undergoes the $P4$ permutation:

P4			
2	4	3	1

Applying the functions, we must perform the following steps: $IP^{-1} \circ f_{K_2} \circ SW \circ f_{K_1} \circ IP$

1. We have already calculated $IP(P) = \{0010\ 0010\}$. Applying the next functions:
2. $f_{K_1}(L, R) = f_{\{1110\ 1001\}}(0010\ 0010) = (0010 \oplus F(0010, \{1110\ 1001\}), 0010)$
3. $F(0010, \{1110\ 1001\}) = P4 \circ SBoxes \circ \{1110\ 1001\} \oplus (E/P(0010))$
4. The steps are:

Bit #	1	2	3	4	5	6	7	8
R	0	0	1	0				
E/P(R)	0	0	0	1	0	1	0	0
k_1	1	1	1	0	1	0	0	1
E/P(R)$\oplus k_1$	1	1	1	1	1	1	0	1
SBoxes(E/P(R)$\oplus k_1$)	1	0	0	0				
P4(Sboxes(E/P(R)$\oplus k_1$))	0	0	0	1				

5. The result from F is therefore 0001
6. Calculating we then have $f_{k_1}(L, R) = (0010 \oplus 0001, 0010) = (0011, 0010)$
7. So far, then $L = 0011$ and $R = 0010$. SW just swaps them so $R = 0011$ and $L = 0010$.
8. We now do the calculation of $f_{k_2}(L, R) = f_{\{1010\ 0111\}}(0010\ 0011) = (0010 \oplus F(0011, \{1010\ 0111\}, 0011))$

9. The steps for F are as above:

Bit #	1	2	3	4	5	6	7	8
R	0	0	1	1				
E/P(R)	1	0	0	1	0	1	1	0
k_2	1	0	1	0	0	1	1	1
E/P(R)$\oplus k_2$	0	0	1	1	0	0	0	1
SBoxes(E/P(R)$\oplus k_2$)	1	0	1	0				
P4(Sboxes(E/P(R)$\oplus k_2$))	0	0	1	1				

10. So now we have the outcome of F as 0011

11. Calculating we then have $f_{k_2}(L, R) = (0010 \oplus 0011, 0011) = (0001, 0011)$

12. Last, we perform the IP^{-1} permutation:

Bit #	1	2	3	4	5	6	7	8
R,L	0	0	0	1	0	0	1	1
IP^{-1}(R,L)	1	0	0	0	1	0	1	0

13. So the final result of the encryption is 1000 1010.