Differential Cryptanalysis

- We deal with input and output differences
- Suppose we know inputs X and X
  - For X the input to S-box is $X \oplus K$
  - For X the input to S-box is $X \oplus K$
  - Key K is unknown
  - **Input difference:** $(X \oplus K) \oplus (X \oplus K) = X \oplus X$
- Input difference is independent of key K
- **Output difference:** $Y \oplus Y$ is (almost) input difference to next round
- Goal is to "chain" differences thru rounds

S-box Differential Analysis

	column			
row	00	01	10	11
0	10	01	11	00
1	00	10	01	11

- ❑ Input diff 000 not interesting
- ❑ Input diff 010 always gives output diff 01
- ❑ More biased, the better (for Trudy)

X ⊕ X	Sbox(X) ⊕ Sbox(X)			
	00	01	10	11
000	8	0	0	0
001	0	0	4	4
010	0	8	0	0
011	0	0	4	4
100	0	0	4	4
101	4	4	0	0
110	0	0	4	4
111	4	4	0	0

Linear Cryptanalysis

- Like differential cryptanalysis, we target the nonlinear part of the cipher
- But instead of differences, we approximate the nonlinearity with **linear equations**
- For DES-like cipher we need to approximate S-boxes by linear functions
- How well can we do this?

S-box Linear Analysis

	column			
row	00	01	10	11
0	10	01	11	00
1	00	10	01	11

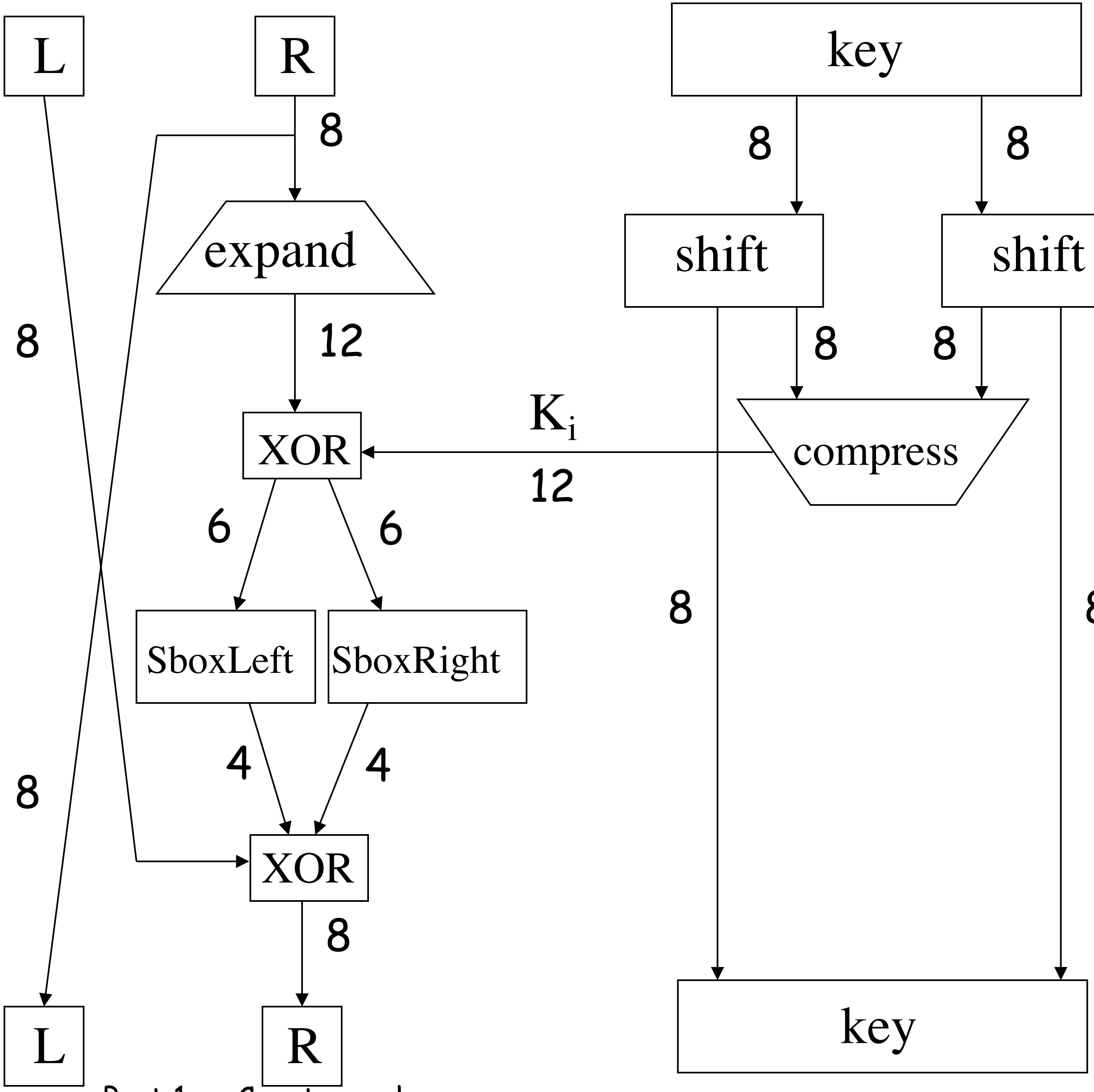
- ❑ Input $x_0x_1x_2$ where x_0 is row and x_1x_2 is column
- ❑ Output y_0y_1
- ❑ Count of 4 is unbiased
- ❑ Count of 0 or 8 is best for Trudy

		output		
		y_0	y_1	$y_0 \oplus y_1$
	0	4	4	4
i	x_0	4	4	4
n	x_1	4	6	2
p	x_2	4	4	4
u	$x_0 \oplus x_1$	4	2	2
t	$x_0 \oplus x_2$	0	4	4
	$x_1 \oplus x_2$	4	6	6
	$x_0 \oplus x_1 \oplus x_2$	4	6	2

Tiny DES (TDES)

- A much simplified version of DES
  - 16 bit block
  - 16 bit key
  - 4 rounds
  - 2 S-boxes, each maps 6 bits to 4 bits
  - 12 bit subkey each round
- Plaintext = (L_0, R_0)
- Ciphertext = (L_4, R_4)

One Round of TDES



Differential Cryptanalysis of TDES

TDES

- TDES SboxRight

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	C	5	0	A	E	7	2	8	D	4	3	9	6	F	1	B
1	1	C	9	6	3	E	B	2	F	8	4	5	D	A	0	7
2	F	A	E	6	D	8	2	4	1	7	9	0	3	5	B	C
3	0	A	3	C	8	2	1	E	9	7	F	6	B	5	D	4

- For X and X suppose $X \oplus X = 001000$
- Then $\text{SboxRight}(X) \oplus \text{SboxRight}(X) = 0010$ with probability 3/4

Differential Crypt. of TDES

- Select P and P so that

  $P \oplus P = 0000\ 0000\ 0000\ 0010 = \mathrm{0x0002}$

- Note that P and P differ in exactly 1 bit
- Let's carefully analyze what happens as these plaintexts are encrypted with TDES

TDES

- Difference of (0000 0010) is expanded by TDES expand perm to diff. (000000 001000)

- If X ⊕ X = 00000010 then F(X, K) ⊕ F(X, K) = 00000010 with prob. ¾

- chain thru multiple rounds

TDES Differential Attack

- Select P and P with $P \oplus P = 0x0002$

$(L_0,R_0) = P$	$(L_0,R_0) = P$	$P \oplus P = 0x0002$
$L_1 = R_0$	$L_1 = R_0$	With probability 3/4
$R_1 = L_0 \oplus F(R_0,K_1)$	$R_1 = L_0 \oplus F(R_0,K_1)$	$(L_1,R_1) \oplus (L_1,R_1) = 0x0202$
$L_2 = R_1$	$L_2 = R_1$	With probability $(3/4)^2$
$R_2 = L_1 \oplus F(R_1,K_2)$	$R_2 = L_1 \oplus F(R_1,K_2)$	$(L_2,R_2) \oplus (L_2,R_2) = 0x0200$
$L_3 = R_2$	$L_3 = R_2$	With probability $(3/4)^2$
$R_3 = L_2 \oplus F(R_2,K_3)$	$R_3 = L_2 \oplus F(R_2,K_3)$	$(L_3,R_3) \oplus (L_3,R_3) = 0x0002$
$L_4 = R_3$	$L_4 = R_3$	With probability $(3/4)^3$
$R_4 = L_3 \oplus F(R_3,K_4)$	$R_4 = L_3 \oplus F(R_3,K_4)$	$(L_4,R_4) \oplus (L_4,R_4) = 0x0202$
$C = (L_4,R_4)$	$C = (L_4,R_4)$	$C \oplus C = 0x0202$

TDES Differential Attack

- Choose P and P with $P \oplus P = 0\text{x}0002$
- If $C \oplus C = 0\text{x}0202$ then

$$R_4 = L_3 \oplus F(R_3, K_4) \qquad R_4 = L_3 \oplus F(R_3, K_4)$$

$$R_4 = L_3 \oplus F(L_4, K_4) \qquad R_4 = L_3 \oplus F(L_4, K_4)$$

  and $(L_3, R_3) \oplus (L_3, R_3) = 0\text{x}0002$

- Then $L_3 = L_3$ and $C=(L_4, R_4)$ and $C=(L_4, R_4)$ are both known
- Since $L_3 = R_4 \oplus F(L_4, K_4)$ and $L_3 = R_4 \oplus F(L_4, K_4)$, for correct choice of subkey K_4 we have

$$R_4 \oplus F(L_4, K_4) = R_4 \oplus F(L_4, K_4)$$

TDES Differential Attack

- Choose P and P with $P \oplus P = \text{0x0002}$
- If $C \oplus C = (L_4, R_4) \oplus (L_4, R_4) = \text{0x0202}$
- Then for the correct subkey K_4

  $R_4 \oplus F(L_4, K_4) = R_4 \oplus F(L_4, K_4)$

  which we rewrite as

  $R_4 \oplus R_4 = F(L_4, K_4) \oplus F(L_4, K_4)$

  where the only unknown is K_4
- Let $L_4 = l_0 l_1 l_2 l_3 l_4 l_5 l_6 l_7$. Then we have

  $$0010 = \text{SBoxRight}(\, l_0 l_2 l_6 l_5 l_0 l_3 \oplus k_{13} k_{14} k_{15} k_9 k_{10} k_{11}) \\ \oplus \text{SBoxRight}(\, l_0 l_2 l_6 l_5 l_0 l_3 \oplus k_{13} k_{14} k_{15} k_9 k_{10} k_{11})$$

TDES Differential Attack

Algorithm to find right 6 bits of subkey K_4

```
count[i] = 0, for i = 0,1,. . .,63
for i = 1 to iterations
    Choose P and P with $P \oplus P$ = 0x0002
    Obtain corresponding C and C
    if $C \oplus C$ = 0x0202
        for K = 0 to 63
            if 0010 == (SBoxRight( $l_0 l_2 l_6 l_5 l_0 l_3 \oplus K$) $\oplus$ SBoxRight( $l_0 l_2 l_6 l_5 l_0 l_3 \oplus K$))
                ++count[K]
            end if
        next K
    end if
next i
```

All K with max count[K] are possible (partial) K_4

TDES Differential Attack

- Experimental results
- Choose 100 pairs P and P with $P \oplus P = 0x0002$
- Found 47 of these give $C \oplus C = 0x0202$
- Tabulated counts for these 47
  - Max count of 47 for each

    $K \in \{000001, 001001, 110000, 111000\}$
  - No other count exceeded 39
- Implies that K_4 is one of 4 values, that is,

  $k_{13}k_{14}k_{15}k_9k_{10}k_{11} \in \{000001, 001001, 110000, 111000\}$
- Actual key is K=1010 1001 1000 0111

Linear Cryptanalysis of TDES

Linear Approx. of Left S-Box

- TDES left S-box or SboxLeft

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	6	9	A	3	4	D	7	8	E	1	2	B	5	C	F	0
1	9	E	B	A	4	5	0	7	8	6	3	2	C	D	1	F
2	8	1	C	2	D	3	E	F	0	9	5	A	4	B	6	7
3	9	0	2	5	A	D	6	E	1	8	B	C	3	4	7	F

- Notation: $y_0y_1y_2y_3 = \text{SboxLeft}(x_0x_1x_2x_3x_4x_5)$
- For this S-box, $y_1=x_2$ and $y_2=x_3$ both with probability 3/4
- Can we "chain" this thru multiple rounds?

TDES Linear Relations

- Recall that the expansion perm is

  $\text{expand}(r_0r_1r_2r_3r_4r_5r_6r_7) = r_4r_7\mathbf{r_2r_1}r_5r_7r_0r_2r_6r_5r_0r_3$

- And $y_0y_1y_2y_3 = \text{SboxLeft}(x_0x_1x_2x_3x_4x_5)$ with $y_1=x_2$ and $y_2=x_3$ each with probability $3/4$
- Also, $\text{expand}(R_{i-1}) \oplus K_i$ is input to Sboxes at round i
- Then $y_1=r_2\oplus k_m$ and $y_2=r_1\oplus k_n$ both with prob $3/4$
- New right half is $y_0y_1y_2y_3\ldots$ plus old left half
- **Bottom line:** New right half bits: $r_1 \leftarrow r_2 \oplus k_m \oplus l_1$ and $r_2 \leftarrow r_1 \oplus k_n \oplus l_2$ both with probability $3/4$

Recall TDES Subkeys

- Key: $K = k_0k_1k_2k_3k_4k_5k_6k_7k_8k_9k_{10}k_{11}k_{12}k_{13}k_{14}k_{15}$
- Subkey $K_1 = k_2k_4\mathbf{k_5k_6}k_7k_1k_{10}k_{11}k_{12}k_{14}k_{15}k_8$
- Subkey $K_2 = k_4k_6\mathbf{k_7k_0}k_1k_3k_{11}k_{12}k_{13}k_{15}k_8k_9$
- Subkey $K_3 = k_6k_0\mathbf{k_1k_2}k_3k_5k_{12}k_{13}k_{14}k_8k_9k_{10}$
- Subkey $K_4 = k_0k_2k_3k_4k_5k_7k_{13}k_{14}k_{15}k_9k_{10}k_{11}$

TDES Linear Cryptanalysis

- Known $P=p_0p_1p_2\ldots p_{15}$ and $C=c_0c_1c_2\ldots c_{15}$

$(L_0,R_0) = (p_0\ldots p_7,p_8\ldots p_{15})$	**Bit 1, Bit 2** (numbering from 0)	**probability**
$L_1 = R_0$	p_9, p_{10}	1
$R_1 = L_0 \oplus F(R_0,K_1)$	$p_1\oplus p_{10}\oplus k_5,\ p_2\oplus p_9\oplus k_6$	3/4
$L_2 = R_1$	$p_1\oplus p_{10}\oplus k_5,\ p_2\oplus p_9\oplus k_6$	3/4
$R_2 = L_1 \oplus F(R_1,K_2)$	$p_2\oplus k_6\oplus k_7,\ p_1\oplus k_5\oplus k_0$	$(3/4)^2$
$L_3 = R_2$	$p_2\oplus k_6\oplus k_7,\ p_1\oplus k_5\oplus k_0$	$(3/4)^2$
$R_3 = L_2 \oplus F(R_2,K_3)$	$p_{10}\oplus k_0\oplus k_1,\ p_9\oplus k_7\oplus k_2$	$(3/4)^3$
$L_4 = R_3$	$p_{10}\oplus k_0\oplus k_1,\ p_9\oplus k_7\oplus k_2$	$(3/4)^3$
$R_4 = L_3 \oplus F(R_3,K_4)$	$k_0 \oplus k_1 = c_1 \oplus p_{10}$	$(3/4)^3$
$C = (L_4,R_4)$	$k_7 \oplus k_2 = c_2 \oplus p_9$	$(3/4)^3$

TDES Linear Cryptanalysis

- Experimental results
- Use 100 known plaintexts, get ciphertexts.
  - Let $P=p_0p_1p_2\ldots p_{15}$ and let $C=c_0c_1c_2\ldots c_{15}$
- Resulting counts
  - $c_1 \oplus p_{10} = 0$ occurs 38 times
  - $c_1 \oplus p_{10} = 1$ occurs 62 times
  - $c_2 \oplus p_9 = 0$ occurs 62 times
  - $c_2 \oplus p_9 = 1$ occurs 38 times
- Conclusions
  - Since $k_0 \oplus k_1 = c_1 \oplus p_{10}$ we have $k_0 \oplus k_1 = 1$
  - Since $k_7 \oplus k_2 = c_2 \oplus p_9$ we have $k_7 \oplus k_2 = 0$
- Actual key is $K = 1010\ 0011\ 0101\ 0110$

To Build a Better Block Cipher…

- How can cryptographers make linear and differential attacks more difficult?
  1. More rounds — success probabilities diminish with each round
  2. Better confusion (S-boxes) — reduce success probability on each round
  3. Better diffusion (permutations) — more difficult to chain thru multiple rounds

Knapsack Lattice Reduction Attack

Lattice?

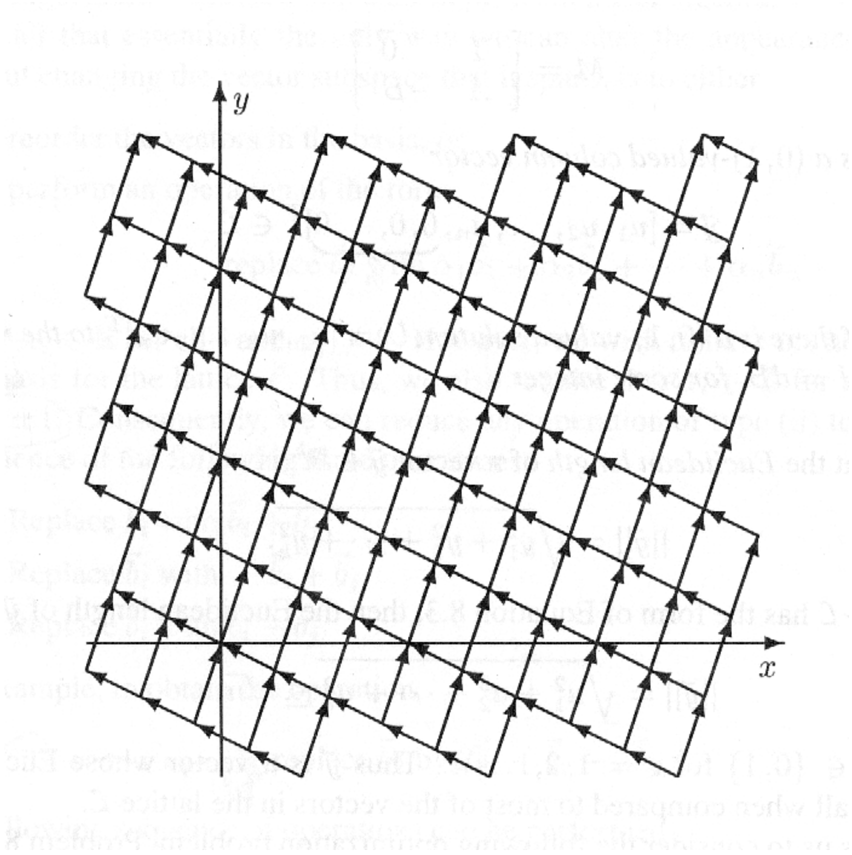
- Many problems can be solved by finding a "short" vector in a **lattice**
- Let $b_1, b_2, \ldots, b_n$ be vectors in $\Re^m$
- All $\alpha_1 b_1 + \alpha_2 b_2 + \ldots + \alpha_n b_n$, each α_i is an integer is a discrete set of points

What is a Lattice?

- Suppose $b_1=[1,3]^T$ and $b_2=[-2,1]^T$
- Then any point in the plane can be written as $\alpha_1 b_1+\alpha_2 b_2$ for some $\alpha_1,\alpha_2 \in \Re$
  - Since b_1 and b_2 are linearly independent
- We say the plane $\Re^2$ is spanned by (b_1,b_2)
- If α_1,α_2 are restricted to **integers**, the resulting span is a lattice
- Then a lattice is a discrete set of points

Lattice Example

- Suppose $b_1=[1,3]^T$ and $b_2=[-2,1]^T$
- The lattice spanned by (b_1,b_2) is pictured to the right



Exact Cover

- **Exact cover** — given a set S and a collection of subsets of S, find a collection of these subsets with each element of S is in exactly one subset
- Exact cover is can be solved by finding a "short" vector in a lattice

Exact Cover Example

- Set $S = \{0,1,2,3,4,5,6\}$
- Spse $m = 7$ elements and $n = 13$ subsets

Subset:	0	1	2	3	4	5	6	7	8	9	10	11	12
Elements:	013	015	024	025	036	124	126	135	146	1	256	345	346

- Find a collection of these subsets with each element of S in exactly one subset
- Could try all 2^{13} possibilities
- If problem is too big, try **heuristic search**
- Many different heuristic search techniques

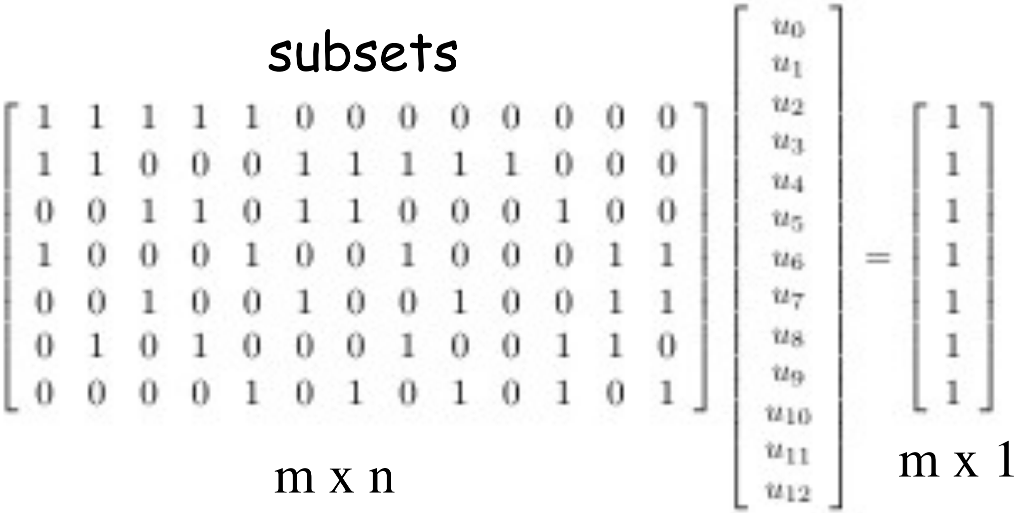
Exact Cover Solution

- Exact cover in matrix form
  - Set $S = \{0,1,2,3,4,5,6\}$
  - Spse $m = 7$ elements and $n = 13$ subsets

Subset:	0	1	2	3	4	5	6	7	8	9	10	11	12
Elements:	013	015	024	025	036	124	126	135	146	1	256	345	346

subsets

elements



$$\begin{bmatrix} 1&1&1&1&1&0&0&0&0&0&0&0&0 \\ 1&1&0&0&0&1&1&1&1&1&0&0&0 \\ 0&0&1&1&0&1&1&0&0&0&1&0&0 \\ 1&0&0&0&1&0&0&1&0&0&0&1&1 \\ 0&0&1&0&0&1&0&0&1&0&0&1&1 \\ 0&1&0&1&0&0&0&1&0&0&1&1&0 \\ 0&0&0&0&1&0&1&0&1&0&1&0&1 \end{bmatrix} \begin{bmatrix} u_0 \\ u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \\ u_7 \\ u_8 \\ u_9 \\ u_{10} \\ u_{11} \\ u_{12} \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

m x n

n x 1

m x 1

Solve: $AU = B$
where $u_i \in \{0,1\}$

Solution:
$U = [0001000001001]^T$

Example

- We can restate AU = B as MV = W where

$$\underbrace{\begin{bmatrix} I_{n\times n} & 0_{n\times 1} \\ A_{m\times n} & -B_{m\times 1} \end{bmatrix}}_{\text{Matrix M}} \underbrace{\begin{bmatrix} U_{n\times 1} \\ 1_{1\times 1} \end{bmatrix}}_{\text{Vector V}} = \underbrace{\begin{bmatrix} U_{n\times 1} \\ 0_{m\times 1} \end{bmatrix}}_{\text{Vector W}} \Longleftrightarrow AU = B$$

- The desired solution is U
  - Columns of M are linearly independent
- Let $c_0, c_1, c_2, \ldots, c_n$ be the columns of M
- Let $v_0, v_1, v_2, \ldots, v_n$ be the elements of V
- Then $W = v_0 c_0 + v_1 c_1 + \ldots + v_n c_n$

Example

- Let L be the lattice spanned by $c_0, c_1, c_2, \ldots, c_n$ (c_i are the columns of M)
- Recall $MV = W$
  - Where $W = [U,0]^T$ and we want to find U
  - But if we find W, we've also solved it!
- Note W is in lattice L since all v_i are integers and $W = v_0c_0 + v_1c_1 + \ldots + v_nc_n$

Facts

- $W = [u_0, u_1, \ldots, u_{n-1}, 0, 0, \ldots, 0] \in L$, each $u_i \in \{0,1\}$
- Then the length of W is

$$\|W\| = \text{sqrt}(u_0^2 + u_1^2 + \ldots + u_{n-1}^2) \leq \text{sqrt}(n)$$

- So W is a very **short** vector in L where
  - First n entries of W all 0 or 1
  - Last m elements of W are all 0
- Can we use these facts to find U?

Lattice Reduction

- If we can find a short vector in L, with first n entries all 0 or 1 and last m entries all 0…
  - Then we *might* have found solution U
- LLL lattice reduction algorithm will efficiently find short vectors in a lattice
- About 30 lines of pseudo-code specify LLL
- No guarantee LLL will find desired vector
- But probability of success is often good

Knapsack Example

- What does lattice reduction have to do with the knapsack cryptosystem?
- Suppose we have
  - Superincreasing knapsack

    $S = [2,3,7,14,30,57,120,251]$
  - Suppose $m = 41, n = 491 \Rightarrow m^{-1} = 12 \bmod n$
  - Public knapsack: $t_i = 41 \cdot s_i \bmod 491$

    $T = [82,123,287,83,248,373,10,471]$
- **Public key:** T **Private key:** (S, m^{-1}, n)

Knapsack Example

- Public key: T Private key: (S, m^{-1}, n)

  $S = [2,3,7,14,30,57,120,251]$

  $T = [82,123,287,83,248,373,10,471]$

  $n = 491, \ m^{-1} = 12$

- Example: 10010110 is encrypted as

  $82+83+373+10 = 548$

- Then receiver computes

  $548 \cdot 12 = 193 \bmod 491$

  and uses S to solve for 10010110

Knapsack LLL Attack

- Attacker knows public key

  $T = [82,123,287,83,248,373,10,471]$

- Attacker knows ciphertext: 548
- Attacker wants to find $u_i \in \{0,1\}$ s.t.

$$82u_0+123u_1+287u_2+83u_3+248u_4+373u_5+10u_6+471u_7=548$$

- This can be written as a matrix equation (dot product): $T \cdot U = 548$

Knapsack LLL Attack

- Attacker knows: $T = [82,123,287,83,248,373,10,471]$
- Wants to solve: $T \cdot U = 548$ where each $u_i \in \{0,1\}$
  - Same form as $AU = B$ on previous slides!
  - We can rewrite problem as $MV = W$ where

$$M = \begin{bmatrix} I_{8\times 8} & 0_{8\times 1} \\ T_{1\times 8} & -C_{1\times 1} \end{bmatrix} = \left[\begin{array}{cccccccc|c} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ \hline 82 & 123 & 287 & 83 & 248 & 373 & 10 & 471 & -548 \end{array}\right]$$

- LLL gives us short vectors in the lattice spanned by the columns of M

LLL Result

- LLL finds short vectors in lattice of M
- Matrix M′ is result of applying LLL to M

$$M' = \begin{array}{c} \begin{array}{ccccccccc} & & & * & & & & & \end{array} \\ \left[\begin{array}{cccccccc|c} -1 & -1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & -1 & 1 & 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & -1 & 1 & 2 \\ 1 & -1 & -1 & 1 & 0 & -1 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & -2 & -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & -1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & -1 \\ \hline 1 & -1 & 1 & 0 & 0 & 1 & -1 & 2 & 0 \end{array}\right] \end{array}$$

- Column marked with "*" has the right form
- Possible solution: $U = [1,0,0,1,0,1,1,0]^T$
- Easy to verify this is actually the plaintext

Side Channel Attack on RSA

Side Channel Attacks

- Sometimes possible to recover key without directly attacking the crypto algorithm
- A **side channel** consists of “incidental info”
- Side channels can arise due to
  - The way that a computation is performed
  - Media used, power consumed, emanations, etc.
- Induced faults can also reveal information
- Side channel may reveal a crypto key

Types of Side Channels

- Emanations security (EMSEC)
  - Electromagnetic field (EMF) from computer screen can allow screen image to be reconstructed at a distance
- Differential power analysis (DPA)
  - Smartcard power usage depends on the computation
- Differential fault analysis (DFA)
- Timing analysis
  - Different computations take different time
  - RSA keys recovered over a network (openSSL)!

The Scenario

- Alice's public key: (N,e)
- Alice's private key: d
- Trudy wants to find d
- Trudy can send any message M to Alice and Alice will respond with M^d mod N
  - That is, Alice signs M and sends result to Trudy
- Trudy can precisely time Alice's computation of M^d mod N

Timing Attack on RSA

- Consider $M^d \bmod N$
- We want to find **private key** d, where $d = d_0d_1\ldots d_n$
- Repeated squaring used for $M^d \bmod N$
- Suppose, for efficiency

```
mod(x,N)
if x >= N
    x = x % N
end if
return x
```

Repeated Squaring

```
x = M
for j = 1 to n
    x = mod(x^2,N)
    if d_j == 1 then
        x = mod(x*M,N)
    end if
next j
return x
```

Timing Attack

- If $d_j = 0$ then
  - $x = \text{mod}(x^2,N)$
- If $d_j = 1$ then
  - $x = \text{mod}(x^2,N)$
  - $x = \text{mod}(x*M,N)$
- Computation time differs in each case
- Can attacker take advantage of this?

Repeated Squaring

```
x = M
for j = 1 to n
    x = mod(x^2,N)
    if d_j == 1 then
        x = mod(x*M,N)
    end if
next j
return x
```

mod(x,N)

```
if x >= N
    x = x % N
end if
return x
```

Timing Attack

- Choose M with $M^3 < N$
- Choose M with $M^2 < N < M^3$
- Let $x = M$ and $x = M$
- Consider $j = 1$
  - $x = \mathrm{mod}(x^2,N)$ does no "%"
  - $x = \mathrm{mod}(x*M,N)$ does no "%"
  - $x = \mathrm{mod}(x^2,N)$ does no "%"
  - $x = \mathrm{mod}(x*M,N)$ does "%" only if $d_1=1$
- If $d_1 = 1$ then $j = 1$ step takes longer for M than for M
- But more than one round…

Repeated Squaring

```
x = M
for j = 1 to n
    x = mod(x², N)
    if d_j == 1 then
        x = mod(x*M, N)
    end if
next j
return x
```

mod(x,N)

```
if x >= N
    x = x % N
end if
return x
```

Timing Attack on RSA

- An example of a chosen plaintext attack
- Choose $M_0, M_1, \ldots, M_{m-1}$ with
  - $M_i^3 < N$ for $i=0,1,\ldots,m-1$
- Let t_i be time to compute $M_i^d \bmod N$
  - $t = (t_0 + t_1 + \ldots + t_{m-1}) / m$
- Choose $M_0, M_1, \ldots, M_{m-1}$ with
  - $M_i^2 < N < M_i^3$ for $i=0,1,\ldots,m-1$
- Let t_i be time to compute $M_i^d \bmod N$
  - $t = (t_0 + t_1 + \ldots + t_{m-1}) / m$
- If $t > t$ then $d_1 = 1$ otherwise $d_1 = 0$
- Once d_1 is known, find d_2 then d_3 then …

Side Channel Attacks

- If crypto is secure Trudy looks for shortcut
- What is good crypto?
  - More than mathematical analysis of algorithms
  - Many other issues (such as side channels) must be considered
- Lesson: Attacker's don't play by the rules!

Crypto Summary

- Terminology, History
- Symmetric key crypto
  - Stream ciphers
    - A5/1 and RC4
  - Block ciphers
    - DES, AES, TEA
    - Modes of operation
    - Integrity

Crypto Summary

- Public key crypto
  - Knapsack
  - RSA
  - Diffie-Hellman
  - ECC
  - PKI, etc.

Crypto Summary

- Hashing
  - Birthday problem
  - Tiger hash
  - HMAC
- Secret sharing
- Random numbers

Crypto Summary

- Information hiding
  - Steganography, Watermarking
- Cryptanalysis
  - Linear and differential cryptanalysis
  - RSA timing attack
  - Knapsack attack