Internet Security [1]
VU 184.216

Engin Kirda engin@infosys.tuwien.ac.at
Christopher Kruegel chris@auto.tuwien.ac.at

News from the Lab

- Challenge 4
  - deadline is next week (31st May)
  - 1/3 of the students have successfully submitted so far
  - we have observed many programming problems
  - please start early
- Challenge 5
  - issued next week (probably on 31st May)
  - deciphering encrypted texts
  - both private and public key schemes

Administration

- DIMVA 2005

  (Detection of Intrusions and Malware & Vulnerability Assessment)
  - security conference co-organized by Engin and myself
  - held in Vienna on 7.-8. July 2005
  - early registration until 2. June 2005
  - student fee is **75 Euro**

- Benefits
  - listen to security research talks given by international experts
  - proceedings book
  - dinner reception at the Rathaus
  - food and gimmicks

- Information and Registration

  http://www.dimva.org/dimva2005/

Cryptography

Cryptography

- (One) definition of cryptography

  Mathematical techniques related to aspects of information security such as

  - confidentiality
    - keep content of information from all but authorized entities
  - integrity
    - protect information from unauthorized alteration
  - authentication
    - identification of data or communicating entities
  - non-repudiation
    - prevent entity from denying previous commitments or actions

History

- Classic cryptography
  - Ancient Egypt
    - non-standard hieroglyphs
  - Hebrew scholars
    - Atbash - mono-alphabetic substitution (reverse of Hebrew alphabet)
  - Greek
    - Steganography (under wax on table, hair of slaves)
  - Roman
    - Caesar cipher - mono-alphabetic substitution (letters are shifted by fixed offset)
  - Alberti (1465)
    - poly-alphabetic substitution

Terminology

- Alphabet of definition A
  - finite set of symbols, e.g., binary alphabet {0,1}
- Message space M
  - set that contains strings from symbols of an alphabet A_1
  - elements of M are called plaintext messages
- Ciphertext space C
  - set that contains strings from symbols of an alphabet A_2
  - elements of C are called ciphertext messages
- Key space K
  - each element $e \in K$ uniquely determines bijective mapping E_e: $M \rightarrow C$ (called encryption function)
  - each element $d \in K$ uniquely determines bijective mapping D_d: $M \rightarrow C$ (called decryption function)

Terminology

- Keys (e,d)
  - not necessarily identical
  - referred to as key pair
- Fundamental
  - all alphabets and the encryption/decryption functions are public knowledge
  - only the selection of the key pair remains secret
- System is breakable
  - if a third party can (without the knowledge of the key pair) systematically recover plaintext from corresponding ciphertext *within some appropriate time frame*
  - exhaustive key search must be made impossible
- Cryptanalysis
  - study of techniques to defeat cryptographic techniques

Taxonomy

- Unkeyed primitives
  - hash functions
  - random sequences
- Symmetric-key primitives
  - block ciphers
  - stream ciphers
  - signatures
  - pseudorandom sequences
- Public-key primitives
  - public-key ciphers
  - signatures

Symmetric-key Cryptography

- Consider an encryption scheme with key pair (e,d)
  - scheme is called a symmetric-key scheme
    if it is “relatively” easy to obtain d when e is know
  - often e = d
- Block cipher
  - break up plaintext into strings (blocks) of fixed length t
  - encrypt one block at a time
  - uses *substitution* and *transposition (permutation)* techniques
- Stream Cipher
  - special case of block cipher with block length t = 1
  - however, substitution technique can change for every block
  - key stream (e_1, e_2, e_3, …)

Block Ciphers

- *Simple (mono-alphabetic) substitution cipher*
  - for each symbol $m_k \in A$ of the plaintext, substitute another symbol $e(m_k)$ according to the permutation p defined by the key e
  - $E_e(m) = (p(m_1), p(m_2), p(m_3), \ldots)$

- Example
  - p: map each letter to the letter three positions on the right in the alphabet

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C

plaintext : `THISC IPHER ISCER TAINL YNOTS ECURE`

ciphertext: `WKLVF LSKHU LVFHU WDLQO BQRWV HFXUH`

Block Ciphers

- *Poly-alphabetic substitution (Vigenere) cipher*
  - for each symbol $m_k \in A$ of the plaintext, substitute another symbol $e(m_k)$ according to one of several permutations p_i defined by the key e
  - for two permutations p_1 and p_2: $E_e(m) = (p_1(m_1), p_2(m_2), p_1(m_3), \ldots)$

- Example
  - using three permutations (mappings)
    - p_1: map to letter that is three positions to the right
    - p_2: map to letter that is seven positions to the right
    - p_3: map to letter that is ten positions to the right

plaintext : `THISC IPHER ISCER TAINL YNOTS ECURE`

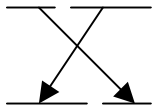
ciphertext: `WOSVJ SSOOU PCFLB WHSQS IQVDV LMXYO`

Block Ciphers

- *Transposition cipher*
  - for each block of symbols $(m_1, \ldots, m_t) \in A$ of the plaintext, the key e defines a permutation on the set {1, …, t } = { p(1), p(2), …, p(t) }
  - $E_e(m) = (m_{p(1)}, m_{p(2)}, \ldots, m_{p(t)},)$

- Example
  - t = 5, permutation is { 3, 4, 5, 1, 2 }

plaintext : `THISC IPHER ISCER TAINL YNOTS ECURE`



ciphertext: `ISCTH HERIP CERIS INLTA OTSYN UREEC`

Block Ciphers

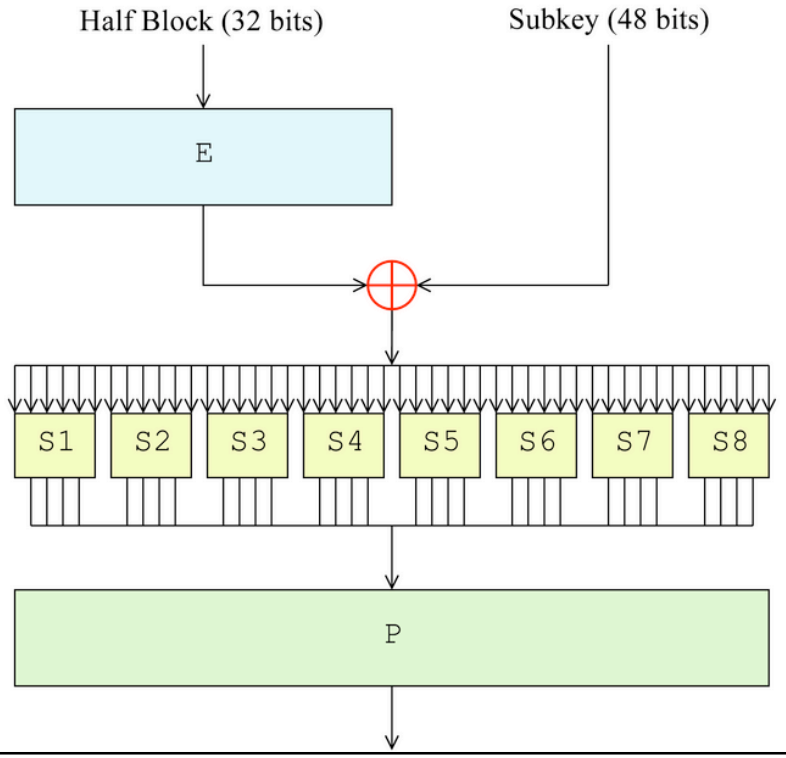
- *Product cipher*
  - combination of substitution and transposition (permutation)
  - often organized in multiple rounds of alternating techniques
    called a SPN (substitution-permutation-network) or Feistel network
  - aims to achieve *confusion* and *diffusion*

- Confusion
  - refers to making the relationship between the key and the ciphertext as complex and involved as possible (achieved via substitution)

- Diffusion
  - refers to the property that redundancy in the statistics of the plaintext is dissipated in the statistics of the ciphertext (via transposition)

Block Ciphers

- Many block ciphers are based on the SPN design
- Data Encryption Standard (DES) is most well-known
  - 64 bit block size
  - 56 bit keys
  - 16 rounds
  - S_1 - S_8
    - S-Boxes
    - non-linear mapping
  - P
    - permutation network



Stream Ciphers

- Block ciphers with t = 1
- $E_e(m) = (e_1(m_1), e_2(m_2), e_1(m_3), \ldots, e_i(m_i))$
- Sequence of keys $e_1, e_2, \ldots, e_i \in K$ is a called a keystream

- Vernam cipher
  - $m_1, m_2, \ldots, m_t \in \{0,1\}$
  - $e_1, e_2, \ldots, e_t \in \{0,1\}$
  - $c_i = m_i \oplus e_i$
  - when e_i are generated randomly and used only once → one-time pad
  - in practice, keystream is often generated from a pseudo-random generator, using a secret seed as the actual key

- RC4
  - used in 802.11 networks for WEP (Wired Equivalent Privacy)

Public-key Cryptography

- Consider an encryption scheme with key pair (e,d)
  - scheme is called a public-key scheme

    if it is computationally infeasible to determine d when e is known

- In public-key schemes, E_e is usually a *trapdoor one-way function* and d is the trapdoor

- One-way function
  - A function f: X → Y is called a trapdoor function, if f(x) is “easy” to compute for all x ∈ X, but for most y ∈ Y, it is infeasible to find a x such that f(x) = y.
  - calculating the exponentiation of an element a in a finite field [a^p (mod n)]
  - multiplication of two large prime numbers [n = p*q]

Public-key Cryptography

- Trapdoor one-way function
  - A trapdoor function f: X → Y with the additional property that given some additional information (called the trapdoor information) it becomes feasible for all y ∈ Y to find a x such that f(x) = y.
- No longer necessary to transfer a secret key over a secure channel
- Significant problem is binding of public key to a certain person (authentication)
  - otherwise, an attacker can substitute his own public key for the victim's key
- Key certificates are needed
  - public key infrastructure (PKI)
  - idea is to cryptographically bind a public key to a certain entity via certificates
  - certificates commonly issued by certification authorities (CAs)
  - chain of trust is traced to a root CA (whose public key must be known by all participants)

RSA

RSA (named after its inventors Rivest, Shamir, and Adleman)

- Suppose user Alice wishes to allow Bob to send her a private message over an insecure transmission medium. She takes the following four steps to generate a public key and a private key:

1. Choose two large prime numbers p, q randomly and independently of each other. Compute $N = p * q$.
2. Compute $\varphi(N) = (p-1)(q-1)$
3. Choose an integer $1 < e < \varphi(N)$ that is coprime to $\varphi(N)$
4. Compute d such that $d * e \equiv 1 \pmod{\varphi(N)}$

- Public key = (e, N)
- Private key = (d, N)
- φ(N) cannot be easily computed from n, but easy from p and q

RSA

The 4 Steps of RSA

1. *Choose two large prime numbers p, q randomly and independently of each other. Compute N = p * q.*

   Can be efficiently done by choosing random numbers of appropriate size and applying fast prime tests.

2. *Compute φ(N) = (p-1)(q-1)*

   Trivial, given p and q.

3. *Choose an integer 1 < e < φ(N) that is coprime to φ(N)*

   Enumerate small prime numbers and check if they divide *φ(N).*

RSA

4. Compute *d* such that $d * e \equiv 1 \pmod{\varphi(N)}$

Can be done using the extended Euclidian algorithm, which calculates the greatest common divisor (gcd) of two numbers a and b (with $a \geq b$)

Rounds	r	q	s	t
0	a	-	1	0
1	b	a / b	0	1
i	$\text{mod}(r_{i-2}, r_{i-1})$	r_{i-1} / r	$s_{i-2} - q_{i-1} * s_{i-1}$	$t_{i-2} - q_{i-1} * t_{i-1}$

- mod(*a*, *b*) is defined as the positive remainder such that $0 \leq \text{mod}(a, b) < b$
- algorithm terminates when $r_{n+1} = 0$

→ then, gcd (a,b) = $r_n = s_n * a + t_n * b$

RSA

Example for extended Euclidian algorithm for a = 23, b = 5

Rounds	r	q	s	t
0	23	-	1	0
1	5	4	0	1
2	3	1	1	-4
3	2	1	-1	5
4	1	2	2	-9
5	0			

gcd (23,5) = 1 = 23 * 2 + (-9) * 5

here is where the magic happens!

→ when gcd (a,b) = 1, then $t_n * b \equiv 1 \pmod{a}$

In our case: $(-9) * 5 \equiv 14 * 5 \equiv 1 \pmod{23}$, and 14 is the inverse of 5 modulo 23

RSA

- Encrypting messages
  - Suppose Bob wishes to send a message *m* to Alice. He turns *m* into a number *n* < *N.* So Bob has *n*, and knows *N* and *e*, which Alice has announced. He then computes the ciphertext *c* corresponding to *n*.

    $$c = n^e \pmod{N}$$

  - e can be large. Nevertheless, the calculation can be done quickly using the method of exponentiation by squaring.

RSA

- Exponentiation by squaring

$$545^{503} \pmod{943} = 545^{256+128+64+32+16+4+2+1} \pmod{943} = 545^{256} \cdot 545^{128} \cdots 545^{1} \pmod{943}$$

$$\begin{aligned}
545^{1} \pmod{943} &= 545 \pmod{943} && = 545 \\
545^{2} \pmod{943} &= 545 \cdot 545 \pmod{943} && = 923 \\
545^{4} \pmod{943} &= 923 \cdot 923 \pmod{943} && = 400 \\
545^{8} \pmod{943} &= 400 \cdot 400 \pmod{943} && = 633 \\
&\cdots \\
545^{256} \pmod{943} &= 18 \cdot 18 \pmod{943} && = 324
\end{aligned}$$

$$545^{503} \pmod{943} = 324 \cdot 18 \cdot 215 \cdot 795 \cdot 857 \cdot 400 \cdot 923 \cdot 545 \pmod{943} = 35 \pmod{943}$$

RSA

- Decrypting messages
  - Alice receives ciphertext c from Bob. She knows her own private key d and can recover the message, which is encoded as n, using

$$n = c^d \pmod{N}$$

- Why does this work?
  - Fermat-Euler theorem: $a^{\varphi(N)} \equiv 1 \pmod{N}$
  - Decoded ciphertext can be written as

$$c^d = (n^e)^d = n^{ed} = n^{1+k\varphi(N)} = n \cdot (n^{\varphi(N)})^k$$

  - Applying the Fermat-Euler theorem yields

$$n \cdot (n^{\varphi(N)})^k \equiv n \cdot (1)^k \equiv n \pmod{N}$$

Cryptanalysis

- Different model (power) of adversary assumed
  - Known-Ciphertext Attack (KCA)
    - you only know the ciphertext
    - requires you know something about the plaintext (e.g., it's English text, an MP3, C source code, …)
    - this is the model for the Sunday cryptograms which use substitution
  - Known-Plaintext Attack (KPA)
    - you have some number of plaintext-ciphertext pairs, but you cannot choose which plaintexts you would like to see
  - Chosen-Plaintext Attack (CPA)
    - you get to submit plaintexts of your choice to an encryption oracle (black box) and receive the ciphertexts in return

Cryptanalysis

- Known-Ciphertext Attack (KCA)
  - weak attack model
  - works only when weak ciphers are used (simple substitution algorithms)

- Attacker can use frequency analysis
  - assumption is that symbols (letters) do not appear with the same frequency in the plaintext
  - this assumption holds with high probability if natural language texts are encrypted
  - in the English language, most frequent letters are E T N R O A S (in this order)

- Attack
  - analyze frequency of symbols in ciphertext
  - assume that symbols with high frequency correspond to frequent letters
  - try to reconstruct plaintext

Cryptanalysis

- Frequency analysis has to be adapted when poly-alphabetic substitution is used
  - in this case, the number of different permutations is most difficult part to find out
  - once the number N of different permutations is known, the ciphertext can be divided into N groups
  - apply frequency analysis individually for each group

- Example with 3 permutations (from the Vigenere cipher)

  plaintext : `THISC IPHER ISCER TAINL YNOTS ECURE`

  ciphertext: `WOSVJ SSOOU PCFLB WHSQS IQVDV LMXYO`

  Group 1: `W, V, S, U, F, W, Q, Q, V, X` | `V(S), W(T), Q(N)`

  Group 2: `O, J, O, P, L, H, S, V, L, Y` | `O(H)`

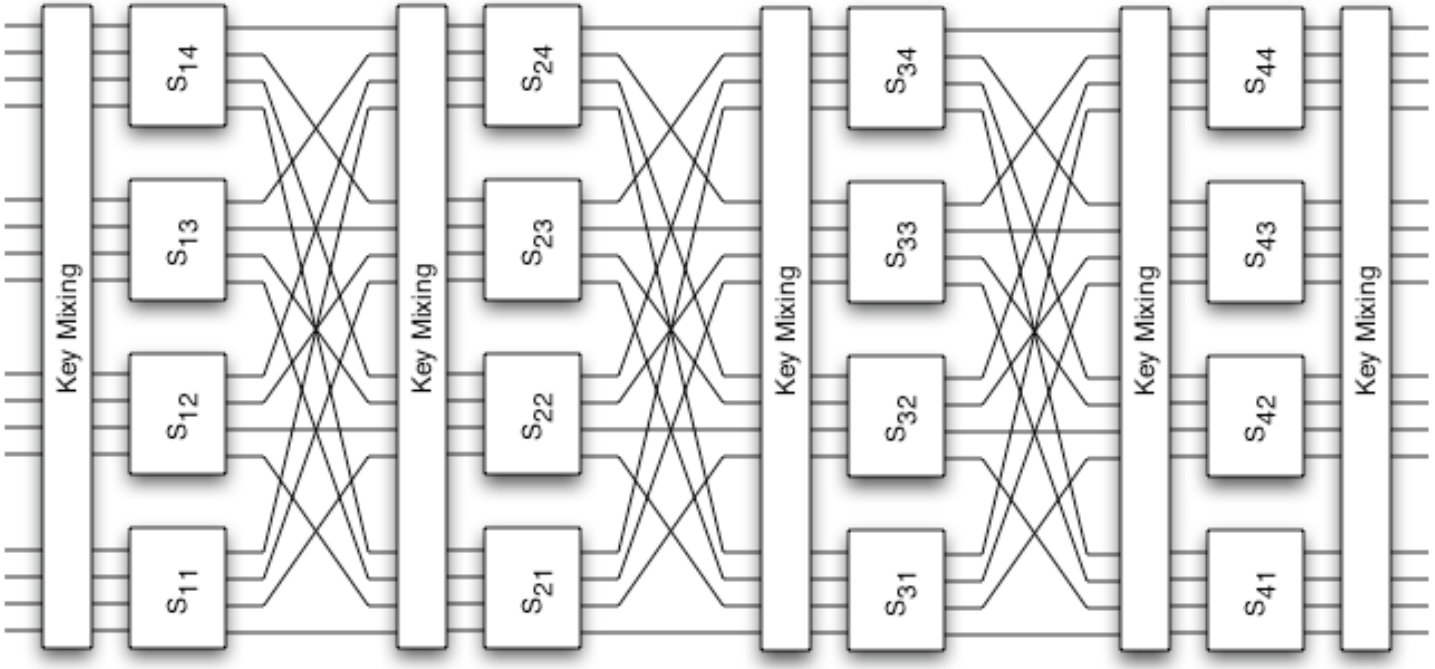
  Group 3: `S, J, O, C, B, S, I, D, M, O` | `S(I), O(E)`

Cryptanalysis

- Better ciphers require more advanced attack techniques
- Two well-known techniques against secret-key block ciphers are
  - *linear cryptanalysis*
    - developed 1993 by Matsui
  - *differential cryptanalysis*
    - discovered three times by NSA, IBM, and Biham and Shamir
- We use a simple four round SPN as example
  - 16 bit key, 16 bit block size
  - S-Box with the following mapping (4 bit input → 4 bit output)

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

Cryptanalysis



Cryptanalysis

- Linear cryptanalysis
  - known plaintext attack
  - exploits high probability occurrences of linear relationships between plaintext, ciphertext, and key bits
  - linear with regards to bitwise operation modulo 2 (i.e., XOR)
  - expressions of form $X_{i1} \oplus X_{i2} \oplus X_{i3} \oplus \ldots \oplus X_{iu} \oplus Y_{j1} \oplus Y_{j2} \oplus \ldots \oplus Y_{jv} = 0$

    X_i = i-th bit of input plaintext [X_1, X_2 , ...]

    Y_j = j-th bit of output ciphertext [Y_1, Y_2, …]

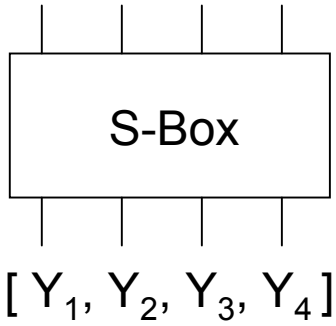
  - for a perfect cipher, such relationships hold with probability 1/2
  - for vulnerable cipher, the probability p might be different from 1/2

  → a bias |p - 1/2| is introduced

Linear Cryptanalysis

- 2 steps
  - analyze the linear vulnerability of a single S-Box
  - connect the output of an S-Box to the input of the S-Box in the next round and “pile up” probability bias

- To analyze a single S-Box, check all possible linear approximations



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

Linear Cryptanalysis

X1	X2	X3	X4	Y1	Y2	Y3	Y4	X1 ⊕ X3 ⊕ X4 = Y2	X2 = Y2 ⊕ Y4
0	0	0	0	1	1	1	0	F	F
0	0	0	1	0	1	0	0	T	F
0	0	1	0	1	1	0	1	T	T
0	0	1	1	0	0	0	1	T	F
0	1	0	0	0	0	1	0	T	F
0	1	0	1	1	1	1	1	T	F
0	1	1	0	1	0	1	1	F	T
0	1	1	1	1	0	0	0	T	F
1	0	0	0	0	0	1	1	F	F
1	0	0	1	1	0	1	0	T	T
1	0	1	0	0	1	1	0	F	F
1	0	1	1	1	1	0	0	T	F
1	1	0	0	0	1	0	1	T	F
1	1	0	1	1	0	0	1	T	T
1	1	1	0	0	0	0	0	T	F
1	1	1	1	0	1	1	1	T	F

Linear Cryptanalysis

- Linear approximations with many true or many false entries are interesting

  $p(X1 \oplus X3 \oplus X4 = Y2) = 12/16 = 0.75$ [bias = 0.25]

  $p(X2 = Y2 \oplus Y4) = 4/16 = 0.25$ [bias = -0.25]

- How to connect probabilities between different rounds?

  consider the following equations, when bias of X1 is b1, and bias of X2 is b2

  $p(X1 \oplus X2 = 0) = p(X1)*p(X2) + (1-p(X1))*(1-p(X2))$

  $= (1/2+b1)*(1/2+b2) + (1/2-b1)*(1/2-b2)$

  $= 1/2 + 2*b1*b2$

Linear Cryptanalysis

- Now, we show how we can eliminate intermediate variables

$p(X1 \oplus X2 = 0) = 1/2 + b1,2$

$p(X2 \oplus X3 = 0) = 1/2 + b2,3$

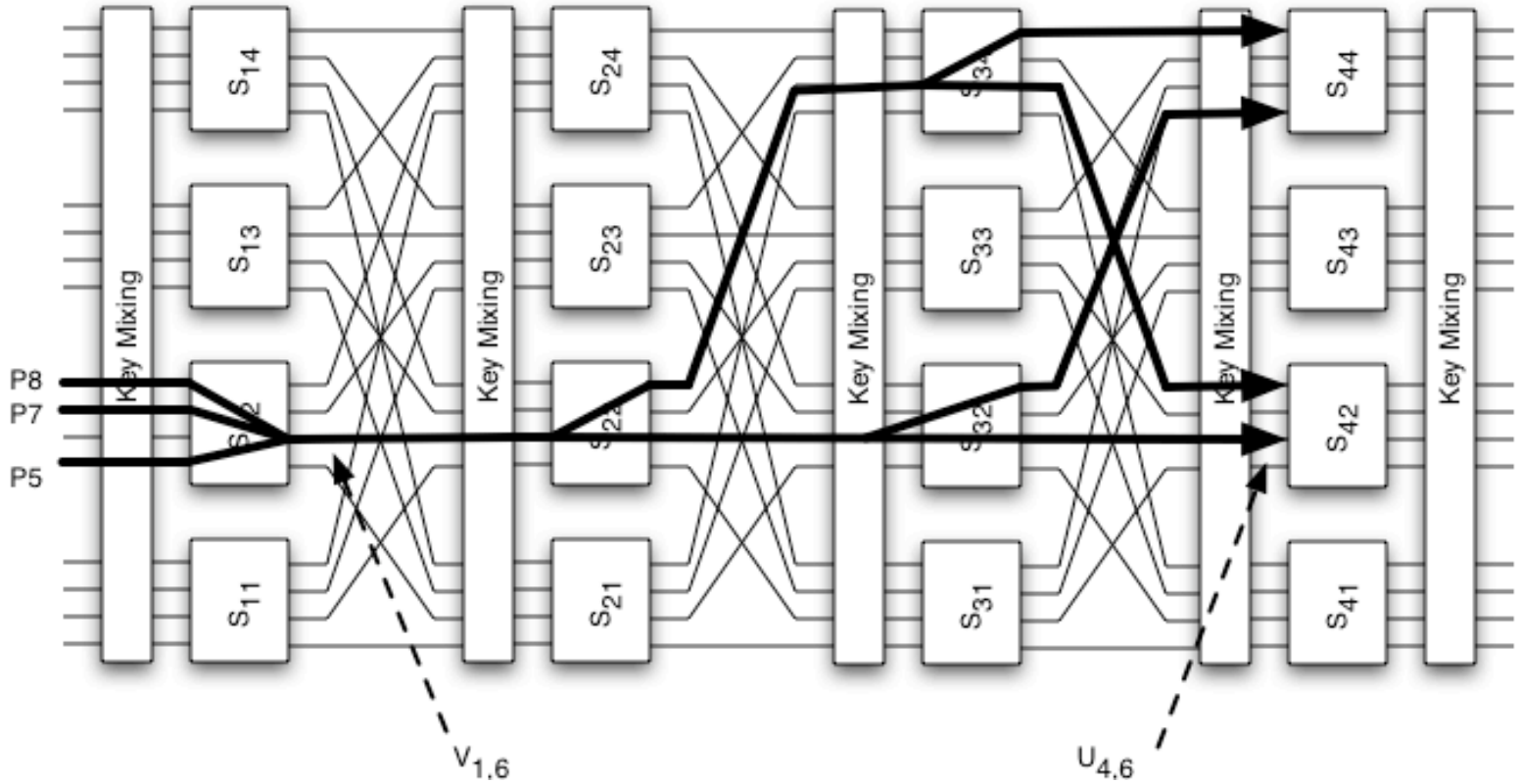
$p(X1 \oplus X3 = 0) = p([X1 \oplus X2] \oplus [X2 \oplus X3] = 0)$

$= p(X1 \oplus X3 = 0)$

$= 1/2 + 2*b1,2 *b2,3$

- Let $U_i(V_i)$ represent the 16-bit block of bits at the input (output) of the S-Box of round i. Then, let $U_{i,k}$ denote the k-th bit of the i-th round of the cipher. Similarly, let K_i represent the key of round i.

Linear Cryptanalysis



Linear Cryptanalysis

- With probability 0.75 (and bias = 0.25), we have

  $V1,6 = U1,5 \oplus U1,7 \oplus U1,8$

  $= (P5 \oplus K1,5) \oplus (P7 \oplus K1,7) \oplus (P8 \oplus K1,8)$

- For the second round, we obtain with probability 0.25 (bias = -0.25)

  $V2,6 \oplus V2,8 = U2,6 \oplus K2,6$

- Because U2,6 = V1,6, we can connect these two equations and get

  $V2,6 \oplus V2,8 = (P5 \oplus K1,5) \oplus (P7 \oplus K1,7) \oplus (P8 \oplus K1,8) \oplus K2,6$

  which can be rewritten as

  $V2,6 \oplus V2,8 \oplus P5 \oplus P7 \oplus P8 \oplus K1,7 \oplus K1,8 \oplus K2,6 = 0$

  This holds with a probability (see before) of 1/2 + 2*0.25*(-0.25) = 0.375

Linear Cryptanalysis

- We continue to eliminate intermediate variables in intermediate rounds to obtain

  $U4,6 \oplus U4,8 \oplus U4,14 \oplus U4,16 \oplus P5 \oplus P7 \oplus P8 \oplus \sum = 0$

  where $\sum$ is a constant factor (either 0 or 1 that depends on a number of key bits)

  This equation holds with a probability of 15/32 (with a bias of -1/32).

  Because $\sum$ is fixed, we know the following linear approximation of the cipher that holds with probability 15/32 or 17/32 (depending on whether $\sum$ is 0 or 1):

  $U4,6 \oplus U4,8 \oplus U4,14 \oplus U4,16 \oplus P5 \oplus P7 \oplus P8 = 0$

Linear Cryptanalysis

- Given an equation that relates the input to the last round of S-Boxes to the plaintext, how can we get the key?
- We attack parts of the key (called target subkey) of the last round, in particular those bits of the key that connect the output of our S-Boxes of interest with the ciphertext

  Given the equation $U4,6 \oplus U4,8 \oplus U4,14 \oplus U4,16 \oplus P5 \oplus P7 \oplus P8 = 0$, we look at the 8 bits K5,5 - K5,8 and K5,13-K5,16

Linear Cryptanalysis

- Idea
  - for a large number of ciphertext and plaintext pairs, we first feed the ciphertext back into the active S-Boxes S_{42} and S_{44}
  - because we do not know the target subkey, we have to repeat this feedback procedure for all possible 256 key
  - for each subkey, we keep a count on how often the linear equation holds
  - when the wrong subkey is used
    - the equation will hold with probability 1/2 (similar to using random values)
  - when the correct subkey is used
    - the equation will hold with more or less often than 1/2 (depending on the bias)

  → after all pairs of plaintext and ciphertext are checked, we take the subkey with the count that differs most from 1/2

Differential Cryptanalysis

- Similar in spirit to linear cryptanalysis
- Chosen plaintext attack
- Instead of linear relationships, sensitivity to modifications of the input are analyzed
  - when certain bits of the input are changed, how does the output change
  - for an ideal cipher, a single bit flip in the input makes all output bits change with a probability of 1/2
  - not always the case
  - probabilistic attack that targets the key of the last round

Conclusion

- Cryptographic schemes
  - symmetric-key cryptography
    - block ciphers
    - DES, SPN, Feistel networks
    - stream ciphers
  - public-key cryptography
    - RSA
- Cryptanalysis
  - frequency analysis
  - linear and differential cryptanalysis

    tutorial on this topic available under http://www.engr.mun.ca/~howard/