

# **An Introduction to Cryptography**

**Edward J. Delp**

**Purdue University**

**School of Electrical and Computer Engineering  
Video and Image Processing Laboratory (*VIPER*)  
West Lafayette, Indiana**

**+1 765 494 1740**

**+1 765 494 0880 (fax)**

**email: [ace@ecn.purdue.edu](mailto:ace@ecn.purdue.edu)**

**<http://www.ece.purdue.edu/~ace>**

**<http://www.ima.umn.edu/~delp>**

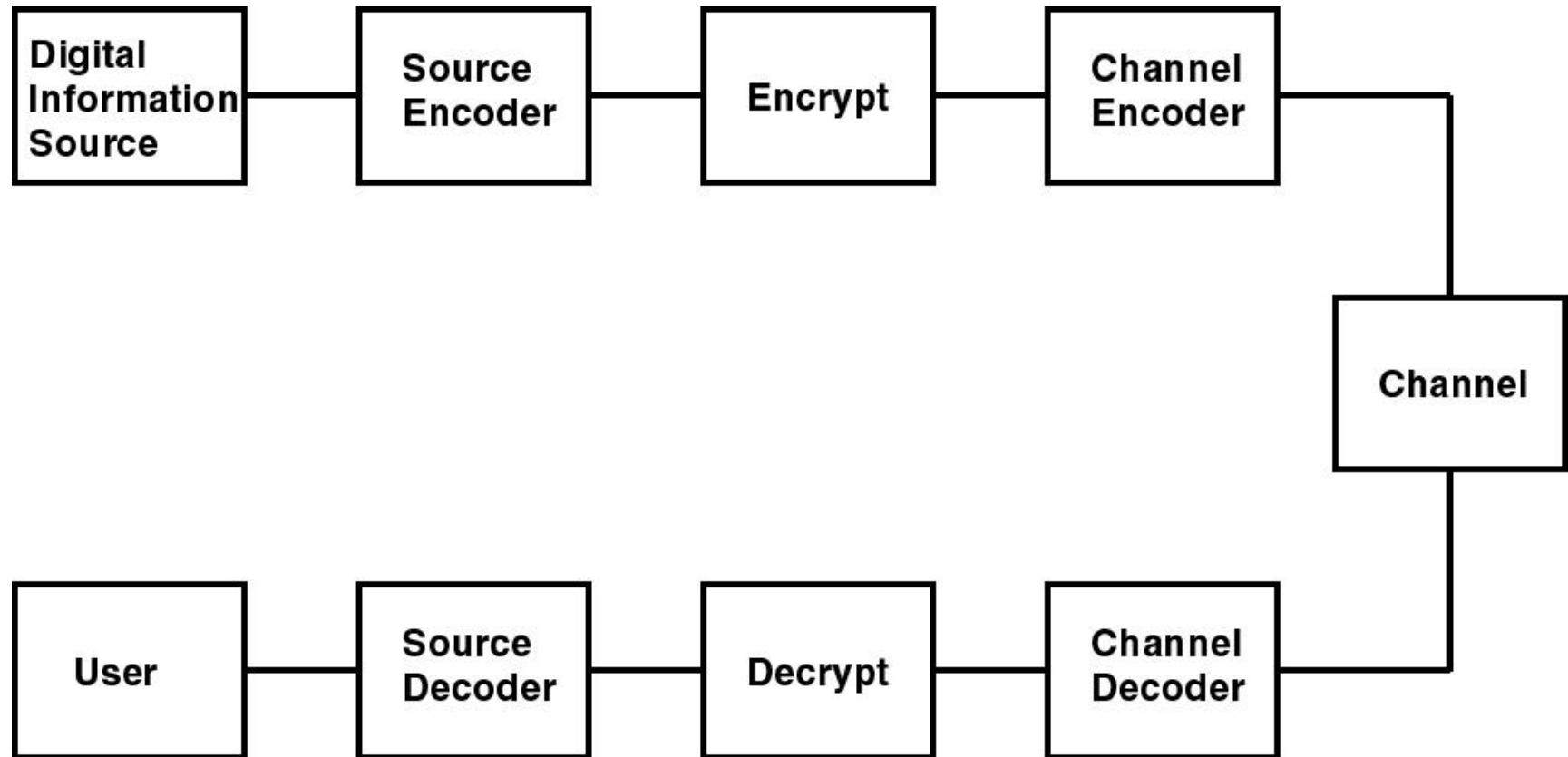


# Outline

- **Provide an introduction to cryptography**



# Digital Communication System



# Cryptography - History

- **Very rich history**
  - **Mary Queen of Scots**
  - **WWII Admiral Yamamoto**
  - **WWII Ultra (Enigma Machine)**
- **“Modern” Cryptography after World War II**
  - **NSA**
- **Popular interest since about 1978**



# Goals

- **Privacy - protect information from unauthorized users**
- **Authentication - “are you who you say you are”**



# Why Is It Now Popular

- **Driven by everything “digital”**
- **Most work to date devoted to text-based or character-based data**



# Export Controls

- **The export of encryption software and hardware is tightly controlled by the US government**
- **Can cause a problem if encryption is included in a product and it is desired to sell it outside the US**



# Cryptography

- **Code** - exploit the linguistic properties of a language
- **Cipher** - do not exploit linguistic properties



# Cryptography



**P - plaintext**  
**C - ciphertext**



# Cryptography

- A special form of computation used to protect a plain-text message
- The “security” of the system is based on the difficulty of the “inverse” computation
- Are there unbreakable ciphers?



# Cryptanalysis

- **Used to break or attack cipher systems**
- **Attack can be brute force (exhaustive search on the keyspace)**
- **Exploit vulnerabilities in the cipher system or the way it is used**
- **“Black bag jobs”**



# Cryptanalysis

- **Known plaintext**
- **Ciphertext only**
- **Chosen plaintext**
- **Cripping**
- **Differential approaches**
- **Traffic flow analysis**
- **Exploit “poor” use of the encryption system**



# Types of Cryptographic Systems



$$C = S(P)$$

$S(\cdot)$  - encryption function

$$P = H(C)$$

$H(\cdot)$  - decryption function



# Types of Cryptographic Systems

- **Totally Secret**
- **Public Algorithm (Secret Key)**
- **Public Key System**



# **Types of Cryptographic Systems**

**Totally secret systems - all aspects of the encryption/decryption is secret, for example “a one time pad”**

**This type of system is very secure but causes programs with managing the use of it**



# Public Algorithm

- **Algorithms are known but parameters are secret**

$$C = S_k(P)$$

$$P = H_k(C)$$

**K » key**

- **Use same key for enciphering and deciphering**
- **Block Ciphers -- DES, IDEA**
- **Stream Ciphers**
- **Problem: key management**



# Public Key Cryptography

- **Two keys**

**E ~ enciphering key**

**D ~ deciphering key**

$$C = S_E(P)$$

$$P = H_D(C)$$

- **Computationally infeasible to derive D from E**
- **Each user could publish E in a “public key directory”**



# Public Key Cryptography

- **No problem with key distribution - really?**
- **Authentication - use private deciphering key to enciphering a message**



# Public Key Cryptography

- **Must protect public key directory**
- **Application of the use of signatures**
- **Certify the public key with a broker of trust (the US Post Office?!)**



# Key Management

- **Block Ciphers - how do you distribute keys**
- **Public Key - protect public key directory**
- **New political issue - key recovery**



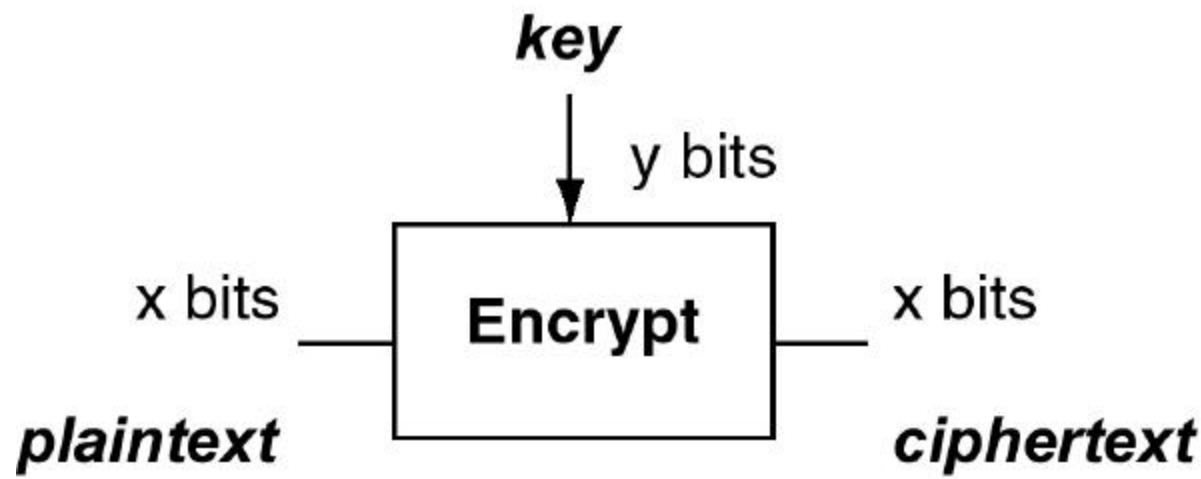
# Public Key Systems

- **Discrete Log (El Gamal)**
- **RSA (Rivest, Shamir, Adleman)**
- **Elliptic Curve Methods**



# Block Ciphers

**Encipher block of  $x$  bits using  $y$  bits of key to produce  $x$  bits of ciphertext**



- **Message extension**
- **Substitution cipher**

# Block Cipher

- **Think of substitution operation as a permutation**
- **$(2^x)!$  Permutations**
- **Key requires  $\log_2[(2^x)!]$  bits**



# Block Ciphers Problems

- **Vulnerable to statistical attacks**
- **Vulnerable to dictionary attacks**



# Feistel Cipher

- Plaintext must be even number of bits,  $2n$
- Plaintext,  $m$ , split into 2 halves  $m = (m_0, m_1)$
- Key has subkeys  $(k_1, k_2, \dots, k_h)$
- Each subkey describes a transformation  $f_{k_i}$  of  $n$  bits into  $n$  bits
- $f_{k_i}$  is a block cipher



# Feistel Cipher

A message  $m$  is enciphered  $h$  times or  $h$  rounds

$$1 \text{ P } \mathbf{u}_0 = (m_0, m_1)$$

$$\mathbf{u}_1 = (m_1, m_2)$$

$$m_2 = m_0 + f_{k_1}(m_1)$$

$$2 \text{ P } \mathbf{u}_1 = (m_1, m_2)$$

$$\mathbf{u}_2 = (m_2, m_3)$$

$$m_3 = m_1 + f_{k_2}(m_2)$$



# Feistel Cipher

$$i^{\text{th}} \text{ P } \mathbf{u}_{i-1} = (m_{i-1}, m_i)$$

$$\mathbf{u}_i = (m_i, m_{i+1})$$

$$m_{i+1} = m_{i-1} + f_{k_i}(m_i)$$

$$h^{\text{th}} \text{ P } \mathbf{u}_{h-1} = (m_{h-1}, m_h)$$

$$\mathbf{u}_h = (m_h, m_{h+1})$$

**Output ciphertext**

$$\mathbf{c} = \mathbf{u}_h$$



# Feistel Cipher

- **Note:**

$$m_{i+1} = m_{i-1} + f_{k_i}(m_i)$$

can also be written as

$$m_{i-1} = m_{i+1} + f_{k_i}(m_i)$$

- **Hence - reverse halves of c and use as input to decipher c**
- **Exact same hardware used for both enciphering and deciphering, i.e do not need  $f_{k_i}^{-1}(\bullet)$**



# **Data Encryption Standard**

## **DES 1977**

- **A Feistel cipher with subkeys that are a function of the round**
- **Based on the IBM Lucifer cipher**
- **A US standard**
- **Several operational modes - block or feedback mode**
- **64-bit plaintext**
- **56-bit key**
- **16 rounds**



# DES

- Input (L, R) (each 32 bits)
- $n^{\text{th}}$  round

input  $L_{n-1}R_{n-1}$

$$L_n = R_{n-1}$$

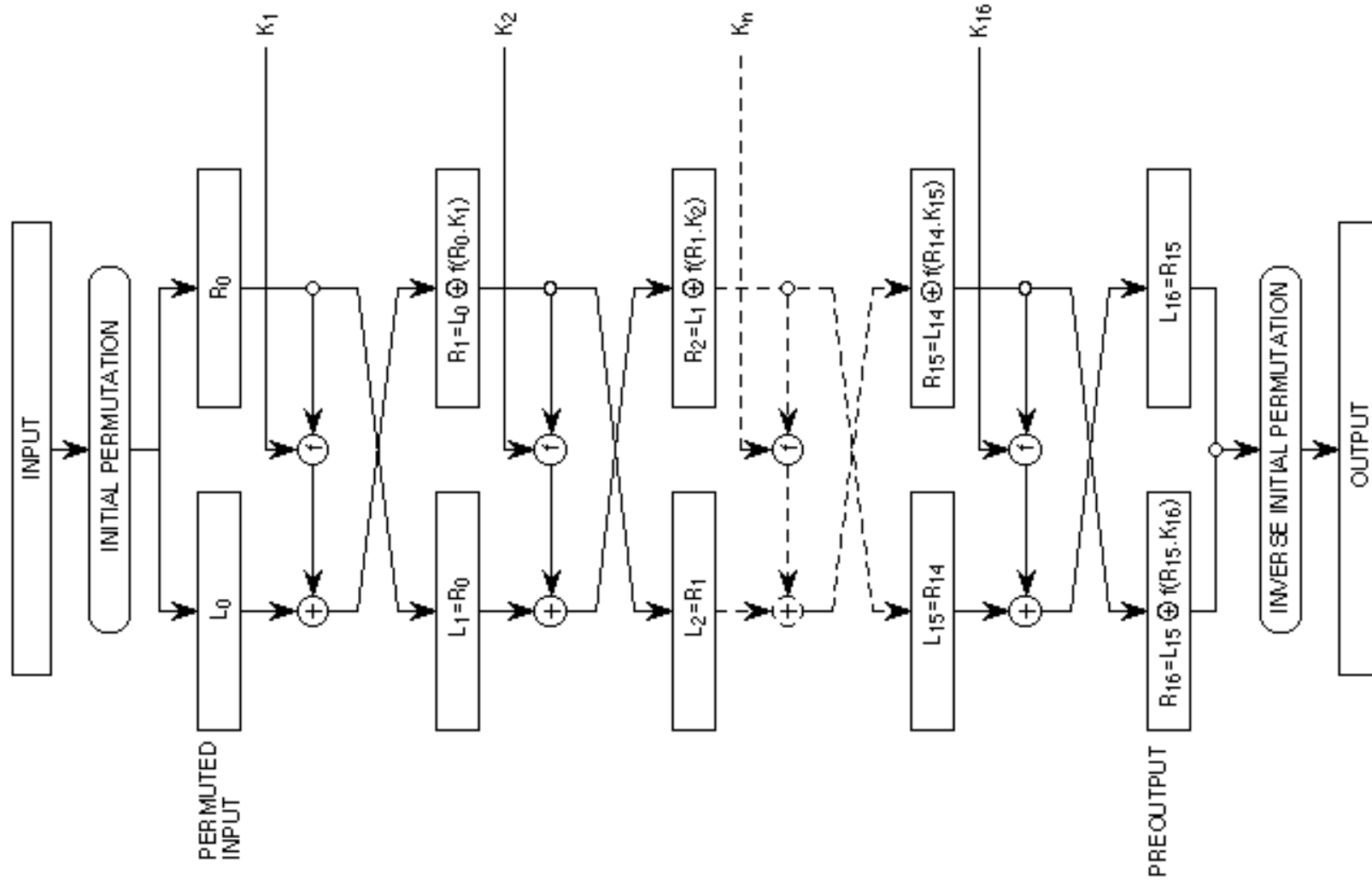
$$R_n = L_{n-1} + f(R_{n-1}, K_n)$$

$K_n \sim 48$  bits chosen for the 56 bit key

$$K_n = \text{KS}(n, \text{key})$$



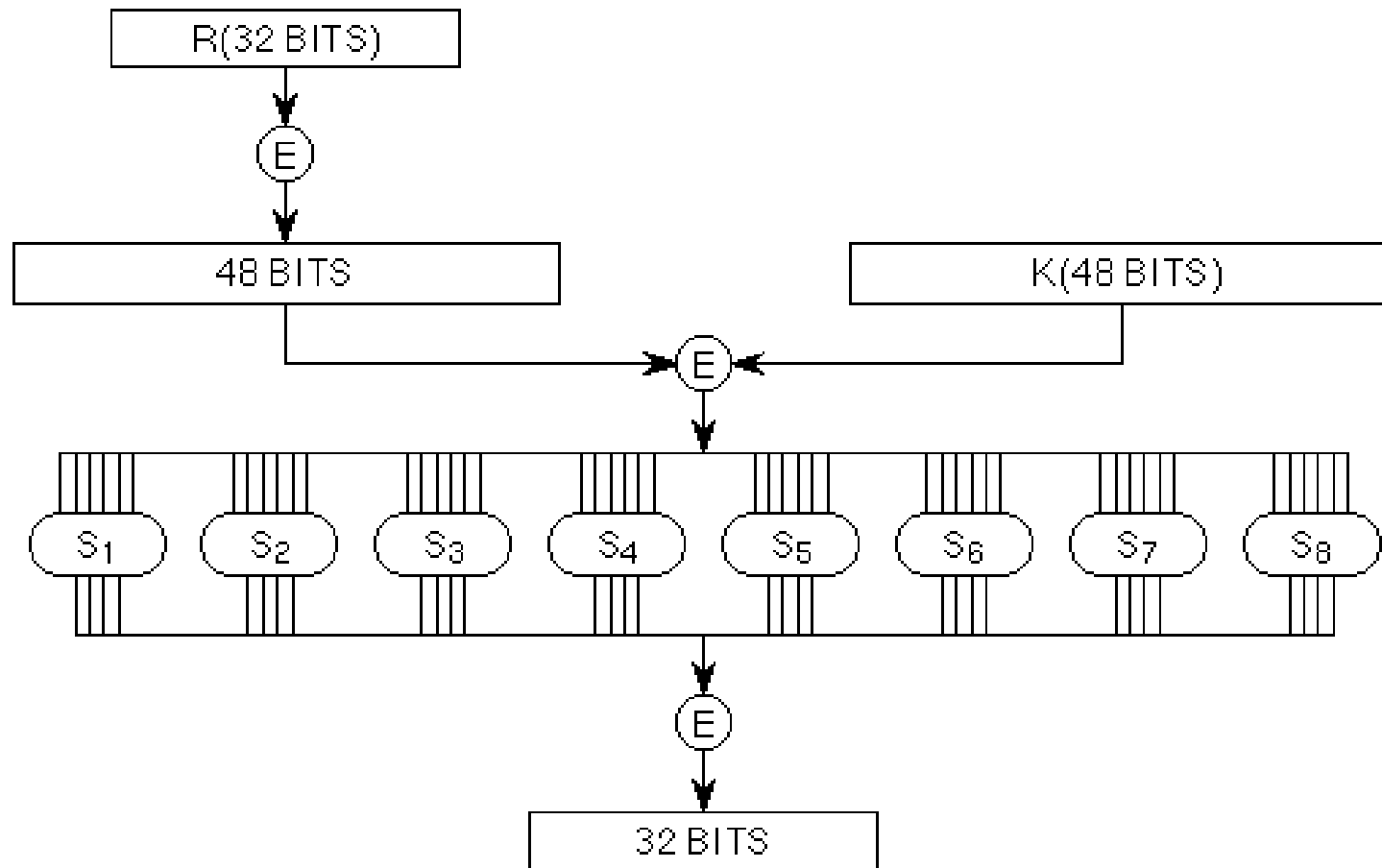
# DES



DES Enciphering computation

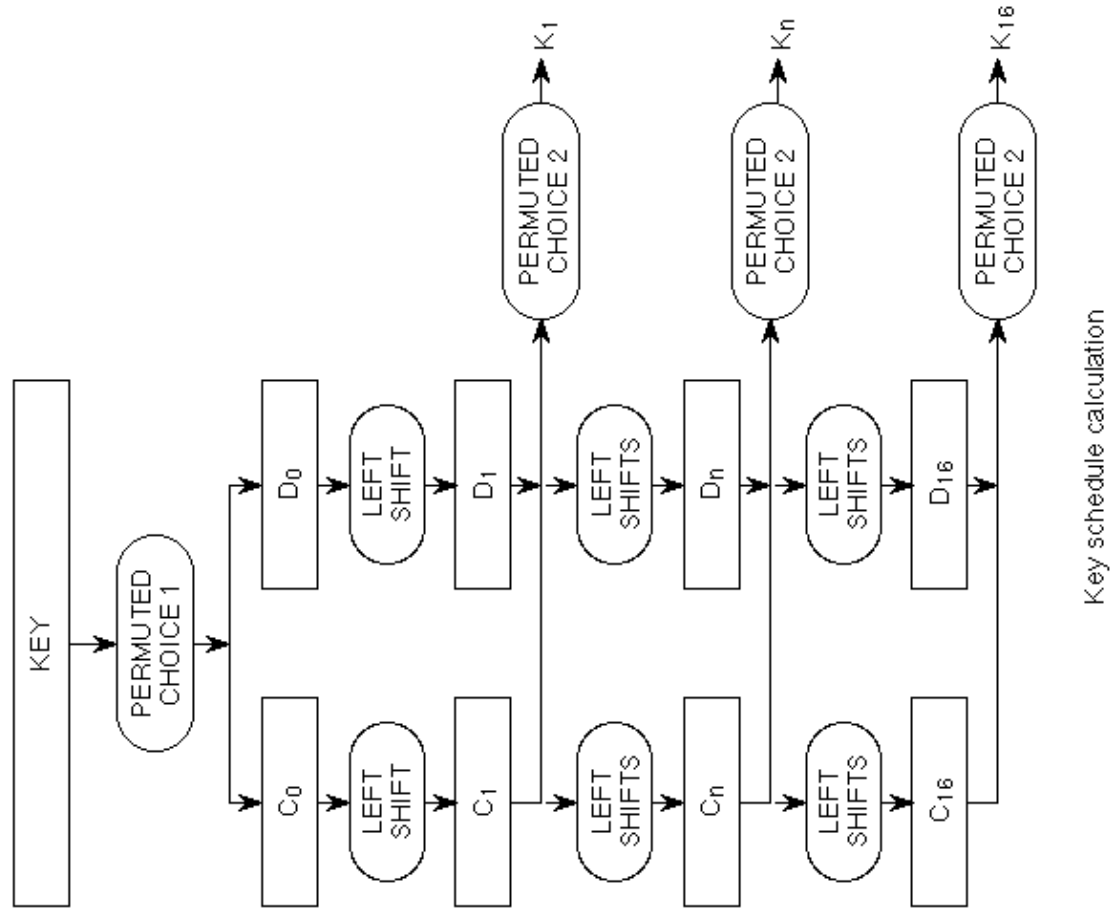


# DES



Calculation of  $f(R, K)$

# DES



# DES

$S_1$

Column Number

Row  
No.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
2	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

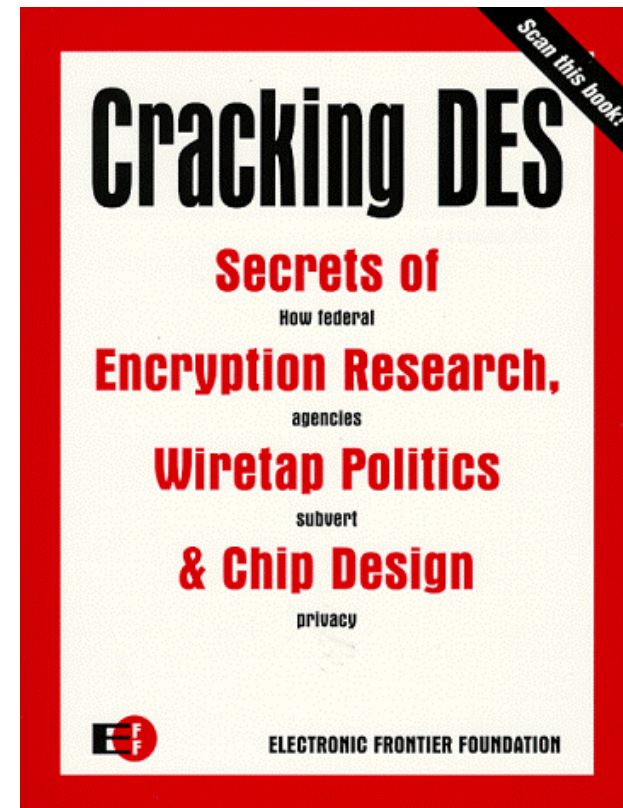


# DES

- **E maps 32-bit input  $\oplus$  48-bit output**
- **S boxes? - 6 bits in / 4 bits out**
  - **MSB and LSB of input form row index**
  - **block ciphers (not affine)**
  - **middle 4 bits form column index**



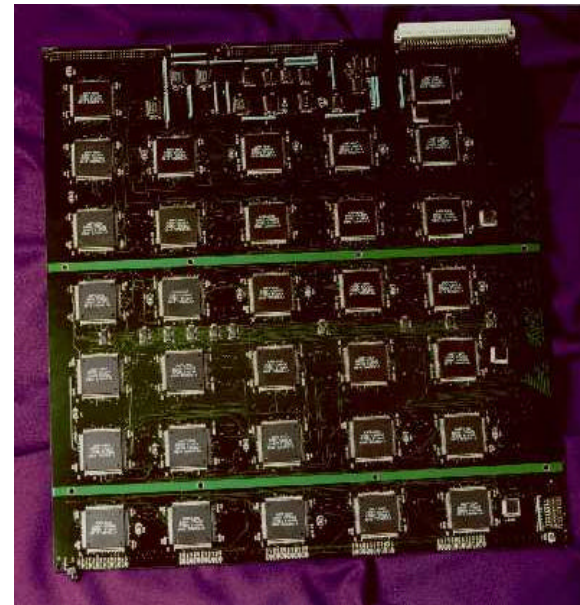
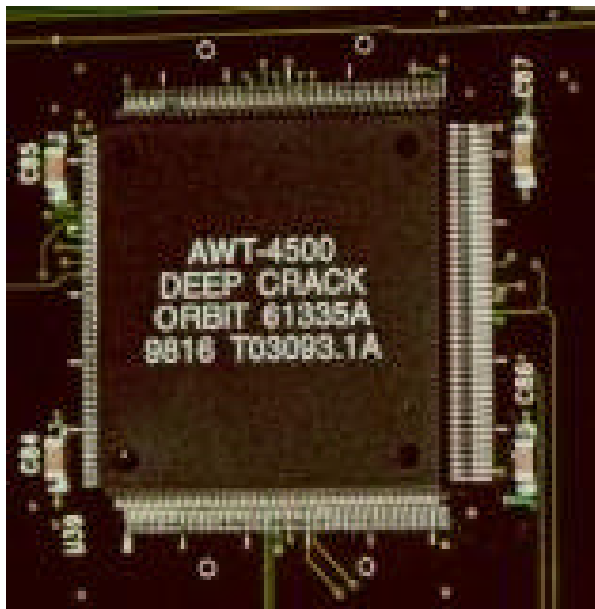
# DES



<http://www.eff.org/descracker.html>



# DES ‘Hardware’



# AES

**Advanced Cryptography Standard - new standard to follow on to DES**

- 128 bit input**
- keys 128, 192, and 256 bits**
- computational requirements**

**New algorithm announced on October 2, 2000**

**It is known as Rijndael**

**<http://csrc.nist.gov/encryption/aes/>**



# Other AES Candidates

- **MARS**
- **RC6**
- **Serpent**
- **Twofish**



# Public Key Cryptography

**RSA (1978)**

**Rivest, Shamir, and Adleman**

**Problem: factor a large integer into the product of two integers**



# RSA

- **Public key: choose integers h and n**
- **Plaintext block: m**
- **Encipher:**  $c = m^h \bmod(n)$
- **Decipher:**  $m = c^d \bmod(n)$
- **h - public enciphering key (known)**
- **d - private deciphering key**
- **n - known**



# RSA

- **Generate d and h - choose two prime numbers p and q such that  $pq = n$**
- **p and q are secret**
- **Choose d such that**

$$\text{GCD}(d, f(n)) = 1$$

$$f(n) = (p-1)(q-1)$$

$f(n) \sim$  **Euler's Totient Function**



# RSA

## Example:

$$p = 61$$

$$q = 53$$

$$n = 3233$$

$$n = 3233$$

$$f(n) = 3120$$

$$\text{choose } d = 37 \text{ } e = 253$$

$$dh = 1 \pmod{f(n)}$$



# RSA

## How to attack RSA

- factor  $n$   $\hat{=}$   $p$  and  $q$   $\hat{=}$   $d$  from  $h$
- $n \sim 300$  digits
  - $\sim 1.5 \times 10^{29}$  operations to factor  $n$
  - 1 ms/operation  $\hat{=}$   $4 \times 10^{15}$  years
- Determine  $f(n)$   $\hat{=}$  factor  $n$



# Public Key Cryptography

## Discrete Log Problem

### El Gamal Cipher

- **p** - prime number
- **a** and **b** integers
- Find **a** such that  $a^a = b \pmod{p}$



# El Gamal Cipher

- **Discrete Log Problem -  $a^a = b \text{ mod } (p)$** 
  - **$p$ ,  $a$ , and  $b$  are public key**
  - **$a$  is secret (deciphering key)**
- **Chose  $k$**
- **$x$  - plaintext**

$$y_1 = a^k \text{ mod } (p)$$

$$y_2 = xb^k \text{ mod}(p)$$

$$c = (y_1, y_2)$$



# El Gamal Cipher

- Plaintext masked by  $b^k$
- decryption - compute  $b^k$  from  $a^k$  and then divide to obtain  $x$

$$x = y_2 (y_1^a)^{-1} \text{ mod } (p)$$

$$y_1^a = (a^k)^a \text{ mod } (p)$$

$$y_1^a = b^k \text{ mod } (p)$$

$$x = x b^k (b^k)^{-1} \text{ mod } (p)$$

- To attack the cipher must solve the discrete log problem for  $a$



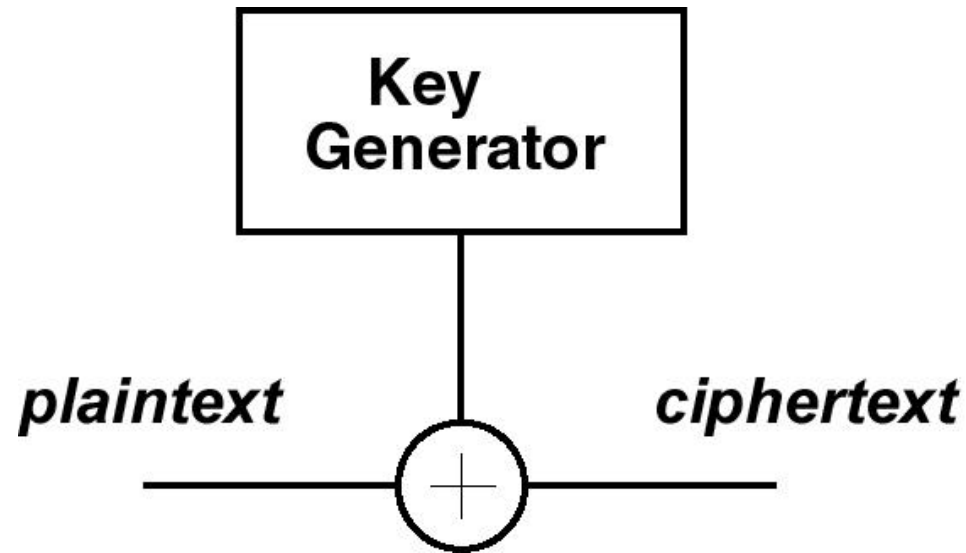
# Diffie-Hellman Key Exchange

- Choose prime number  $n$  and integer  $g$  - can be made public
- User 1  $\mathcal{P}$   $A = g^x \bmod n$  ( $x$  random integer); send  $A$  to User 2
- User 2  $\mathcal{P}$   $B = g^y \bmod n$  ( $y$  random integer): send  $B$  to User 1
- User 1  $\mathcal{P}$   $k = B^x \bmod n$
- User 2  $\mathcal{P}$   $h = A^y \bmod n$
- $k = h = g^{xy}$  use as the key

illegal user knows:  $n$ ,  $g$ ,  $A$ , and  $B$   $\mathcal{P}$  to find key - solve the discrete log



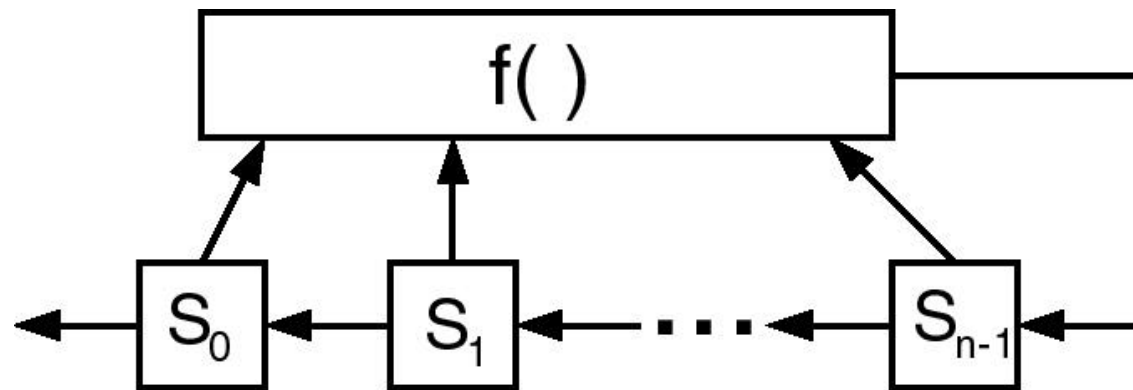
# Stream Ciphers



- **Key generator - generate random sequence**
- **Can it be random?**

# Key Generator

- **Shift Register sequence**



- **Linear Shift Register Sequence**

# Authentication Signature Schemes

- **Who are you?**
- **Are you who you say you are?**
- **Signing a document**



# Signatures

- **Digital Signatures vs. Conventional Signatures**
- **“Signing” a document**
  - **Conventional Signature - physically part of the document**
  - **Digital Signature - must have a “binding” operation to bind signature to message**
- **Verification**
  - **Conventional - compare to other authentic document**
  - **Digital - public algorithm anyone can verify the signature**



# Signatures

- **A copy of signed digital document is identical to the original**
- **Problem with document reuse (time-stamping)**



# Signature Algorithm

- **Signing Algorithm**  $\text{sig}_k(\bullet)$
- **Verification Algorithm**  $\text{ver}_k(\bullet)$
- **El Gamal Signature Algorithm**
- **DSS (December 1, 1994)**
- **Difference in Encryption and Signature Systems**
  - **Signature System must be stronger**
  - **Problems with signing long messages**



# Hash Functions

- **Hash functions convert arbitrary-length binary strings to a fixed length output,  $H = H(P)$**
- **Useful properties:**
  - **trivial to produce  $H$ , given  $P$**
  - **extremely difficult to obtain  $P$  from  $H$**
  - **very difficult to find two inputs,  $P_1$  and  $P_2$ , that yield the same  $H$  (collision resistance)**



# Hash Functions

- **Produce Message Digest by “hashing” the message**
  - check sum
  - map large message into n bit hash
- **Sign message digest**
- **MD4 Hash (Rivest 1990)**
- **MD5 Hash (Rivest 1991) 128 bit hash**
- **Secure Hash Standard (SHS) (May 11, 1993) 160 bit hash**
- **SHA-1**



# Time Stamps

- Time stamps use hash functions to verify a digital work's time of creation, ownership and content:
  - When was this data created or last modified?
- Two procedures:
  - certification - the author of the data can "sign" the record, or a user can fix data in time. The result is a certificate
  - verification - *any* user can check data and its certificate to make sure it is correct
- Time stamping is a form of authentication and requires a "trusted" third party escrow agent
- <http://www.surety.com/>



# Pretty Good Privacy - PGP

- Uses RSA, IDEA, and MD5 hash
- Message encrypted using IDEA
  - 64 bit plaintext, 128 bit key
- RSA used to encrypt IDEA key
- Hash used for signing

<http://www.pgp.com/>



# Certificates and Digital I.D.

- **Use to certify that your public key is correct - trusted third party signs your public key and issues a certificate or “digital I.D”**
- **Used**
  - **web browsers**
  - **secure email**
  - **smart cards**



# **Certification Authority (Trusted Agents)**

- **VeriSign - [www.verisign.com](http://www.verisign.com)**
- **GTE CyberTrust Solutions -  
[www.bbn.com/products/security/cytrust/index2.htm](http://www.bbn.com/products/security/cytrust/index2.htm)**
- **Entrust - [www.entrust.com](http://www.entrust.com)**

**All use the Public Key Infrastructure (PKI)**

**<http://csrc.nist.gov/pki/>**



# **Digital Millennium Copyright Act**

**Will it be illegal to remove security features from a data element?**

**<http://lcweb.loc.gov/copyright/>**

**<http://www.dfc.org/>**



# Reference Books

- **D. R. Stinson, *Cryptography: Theory and Practice*, CRC Press, 1995.**
- **B. Schneier, *Applied Cryptography, (2nd edition)* Wiley, 1996.**
- **D. Kahn, *The Codebreakers*, Scribner, 1996.**
- **K. W. Dam and H. S. Lin, *Cryptography's Role In Securing The Information Society*, National Academy Press, 1996.**



# Web Resources

- **RSA Data Security -- <http://www.rsa.com> (excellent FAQ)**
- **International Association for Cryptologic Research  
<http://www.swcp.com/~iacr>**
- **Ron Rivest's Cryptography and Security Page  
<http://theory.lcs.mit.edu/~rivest/crypto-security.html>**
- **Trusted Information Systems - <http://www.tis.com>**
- **Dorothy Denning's Cryptography Project  
<http://www.cosc.georgetown.edu/~denning/crypto>**
- **Bruce Schneier's Counterpane  
<http://www.counterpane.com/>**

