Quo Vadis Cryptology?

Josef Pieprzyk
School of Electrical Engineering and Computer Science
QUT, Brisbane, Australia
&
Instytut Podstaw Informatyki,
PAN, Warszawa, Polska

January, 2016

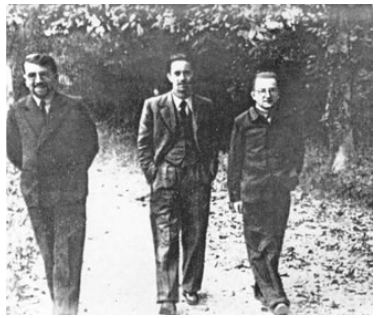
Queensland University of Technology

Outline

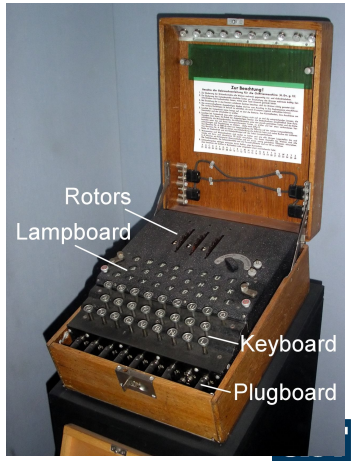
1. Private-key Cryptography
2. Public-key Cryptography
3. Multiparty Computations
4. Cryptanalysis

Queensland University of Technology

PART 1 – Private-Key Cryptography



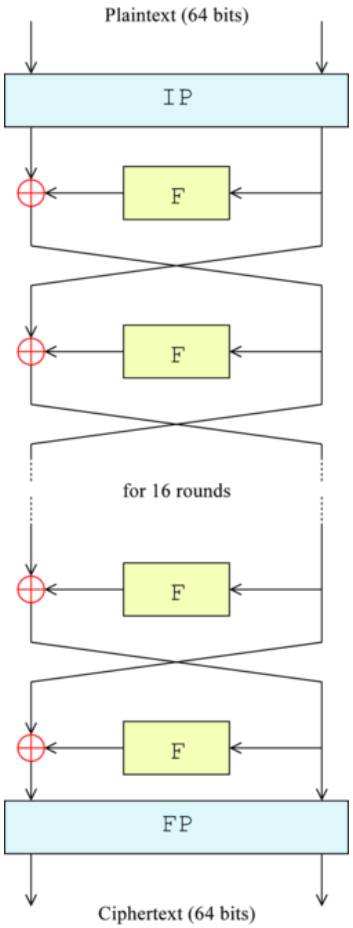
- Military cryptography (Enigma) - an example of early encryption machines

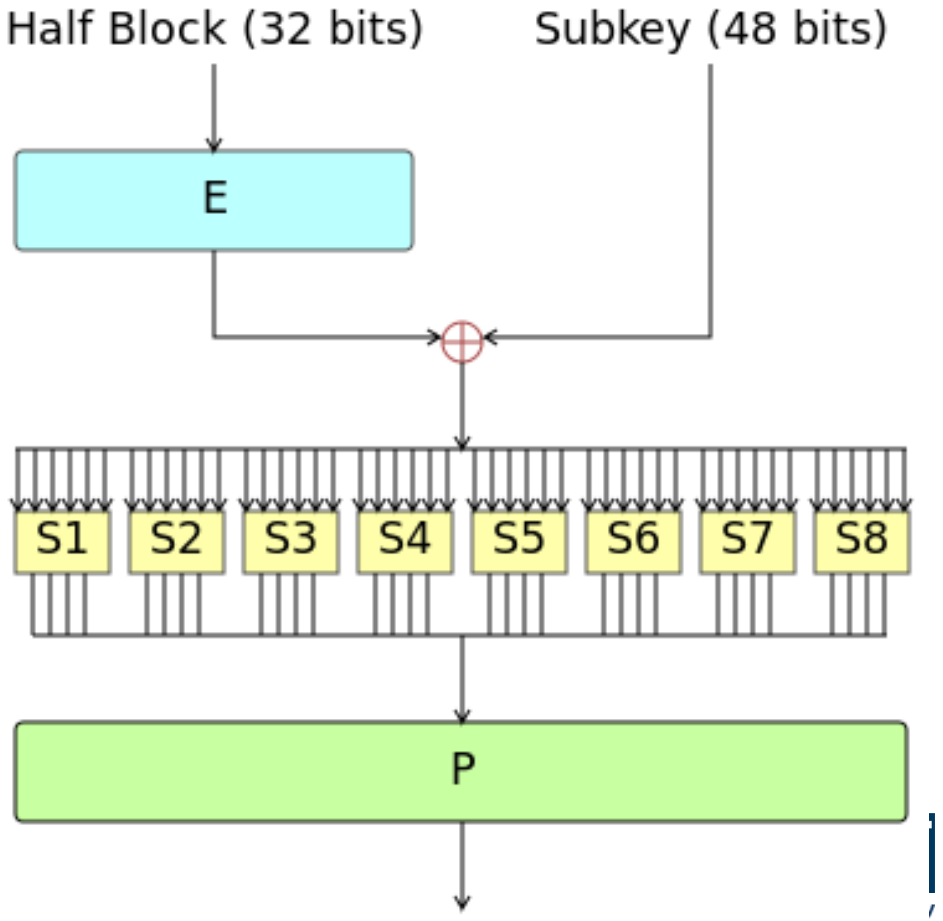


Queensland University of Technology

PART 1 – DES

- Data Encryption Standard (1975) - block cipher for non-military applications (IBM) – NIST Standard
- Feistel structure, 4 $\times$ 6 eight S-boxes
- 56-bit keys

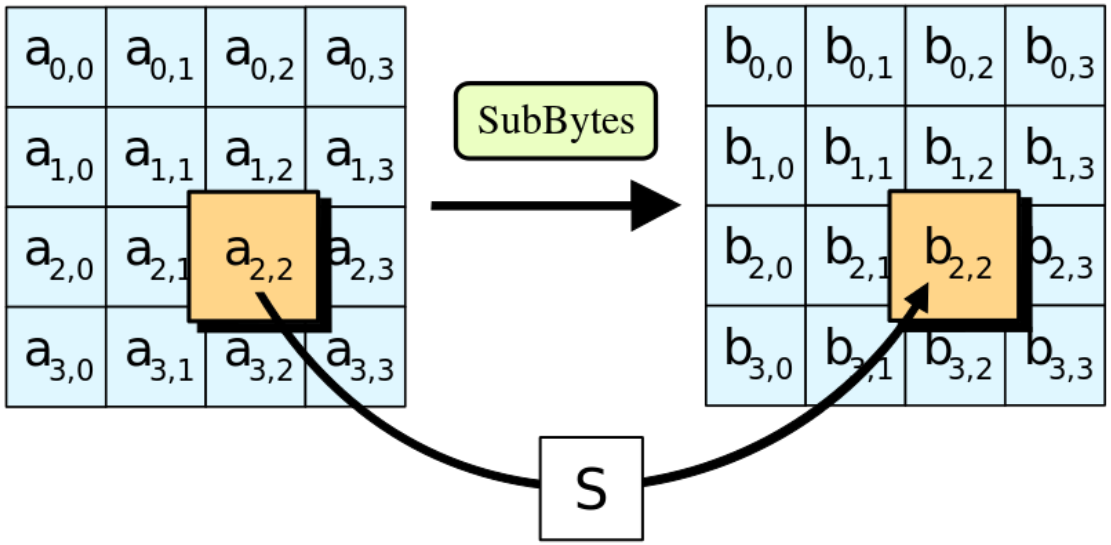




PART 1 – AES

Advanced Encryption Standard

- Public AES competition announced by NIST in 1997
- Finalists: Rijndael, Serpent, Twofish, RC6, MARS
- Winner - Rijndael (Vincent Rijmen and Joan Daemen) - 2001
- SP network structure, 8 $\times$ 8 S-box

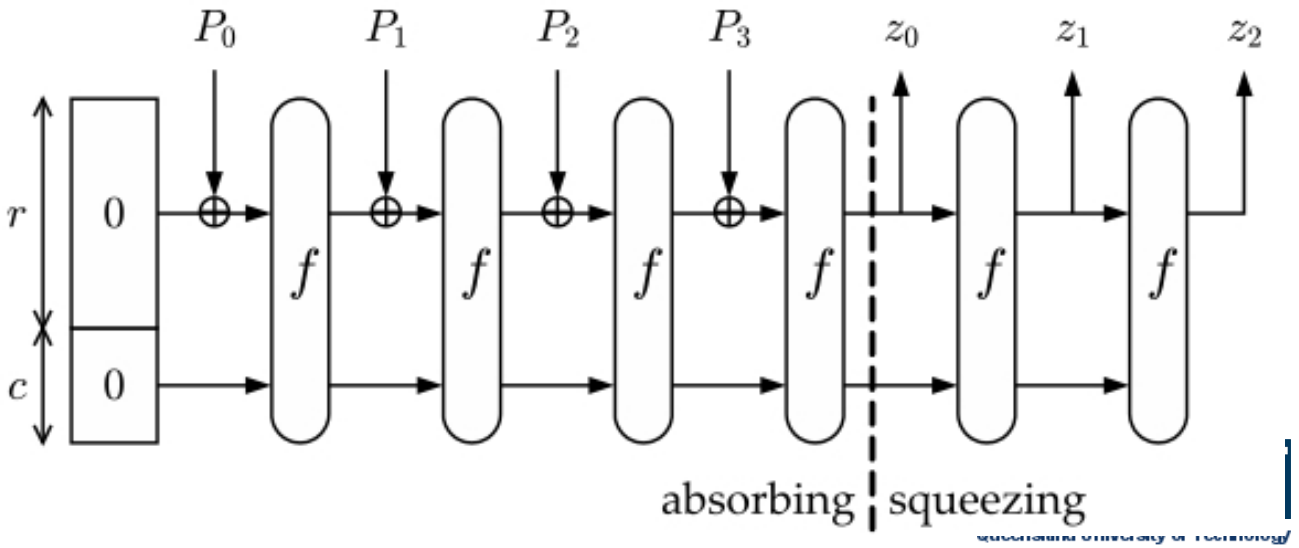


Queensland University of Technology

PART 1 – SHA3

Secure Hash Algorithm Standard

- Cryptanalysis by Xiaoyung Wang
- SHA3 Competition - NIST 2007
- Finalists: Blake, Grøstl, JH, Keccak and Skein
- Winner - Keccak, 2012 (Guido Bertoni, Joan Daemen, Michael Peeters and Gilles Van Assche)
- Sponge structure



PART 1 – CAESAR

Authenticated Encryption Competition (Daniel Bernstein)

- Deadline for submissions March 2014
- 57 submissions (AES based, Sponge structures, stream cipher based, hash function based)
- July 2015 – 2nd Round – 29 candidates
- March 2016 – 3rd Round
- December 2016 - announcement of finalists
- December 2017 - announcement of winner

Queensland University of Technology

Outline

1 Private-key Cryptography

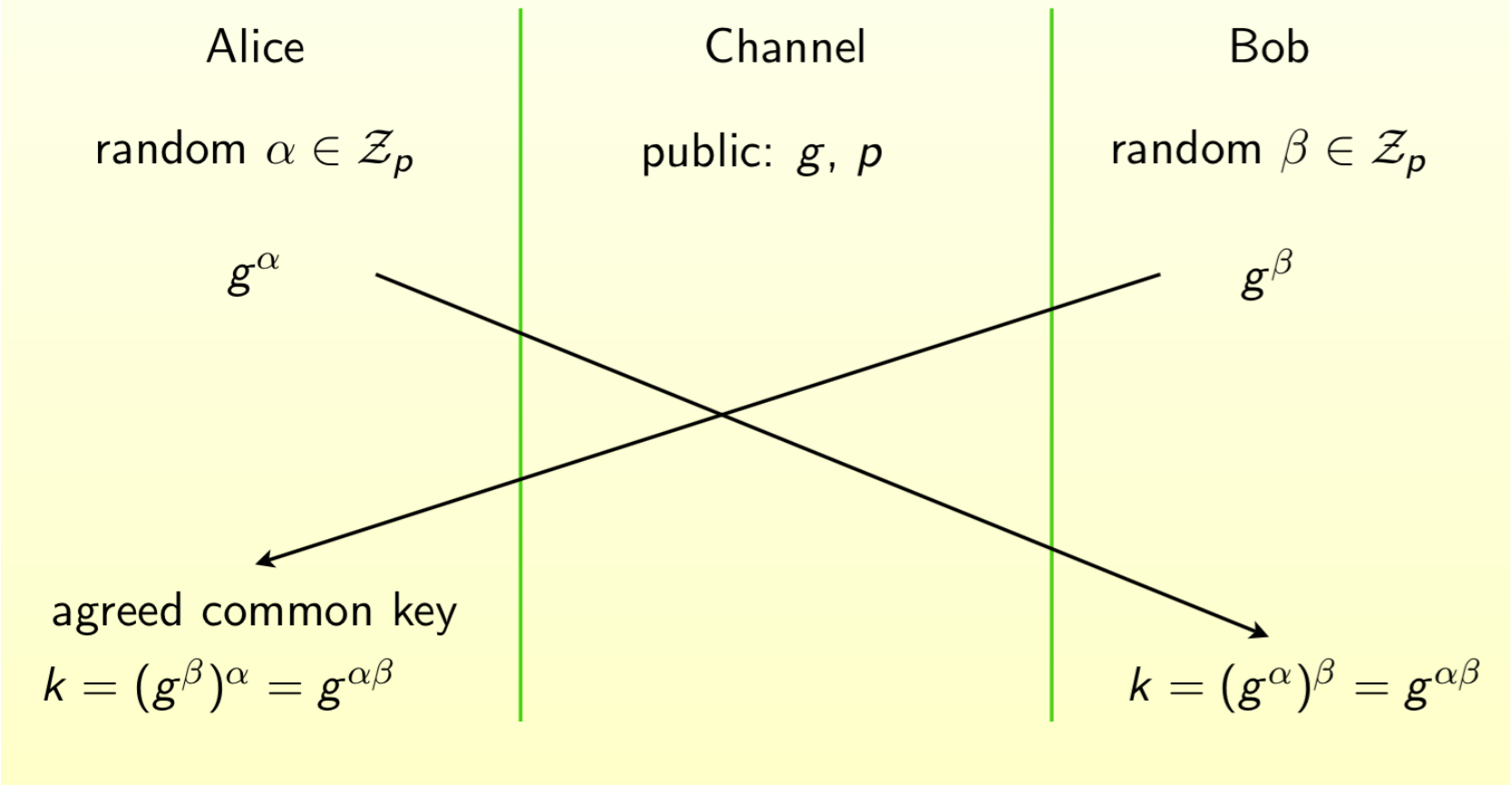
2 Public-key Cryptography

3 Multiparty Computations

4 Cryptanalysis

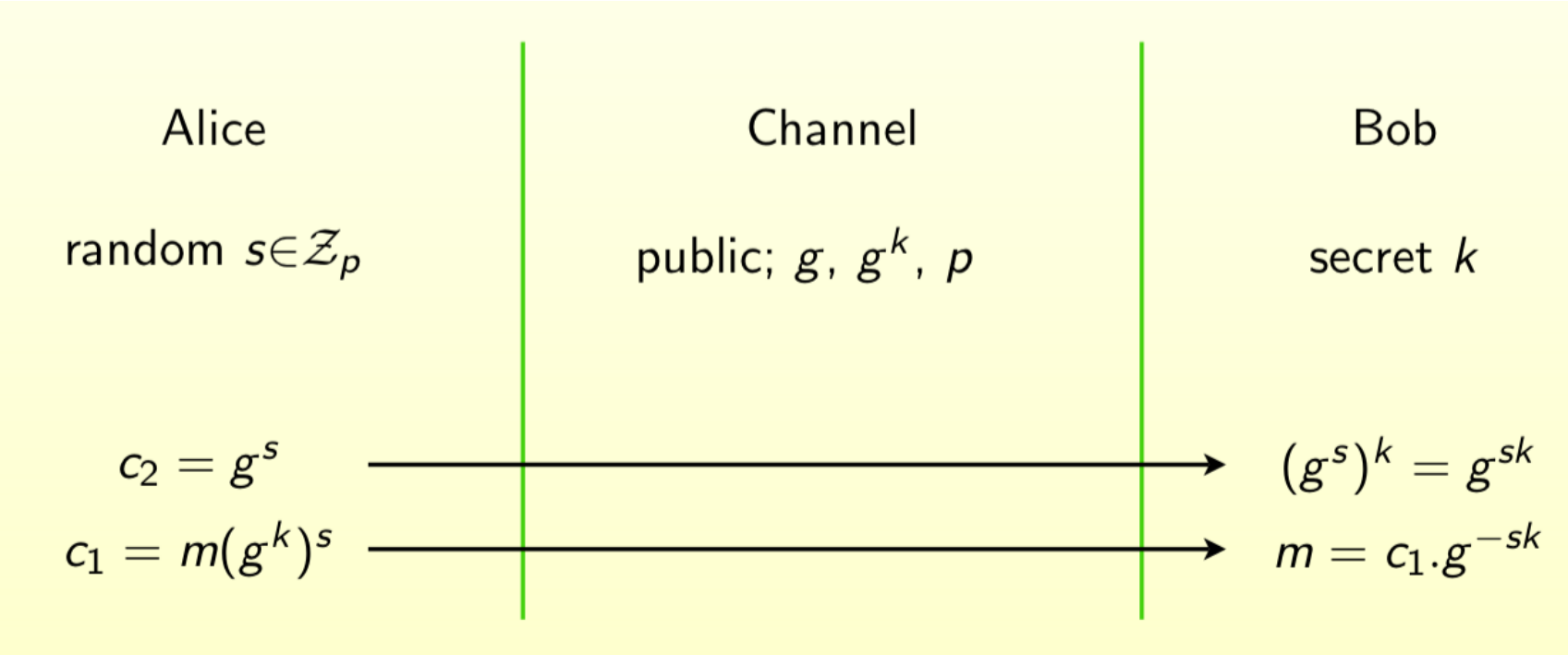
Queensland University of Technology

PART 2 – Diffie-Hellman Key Agreement (1976)



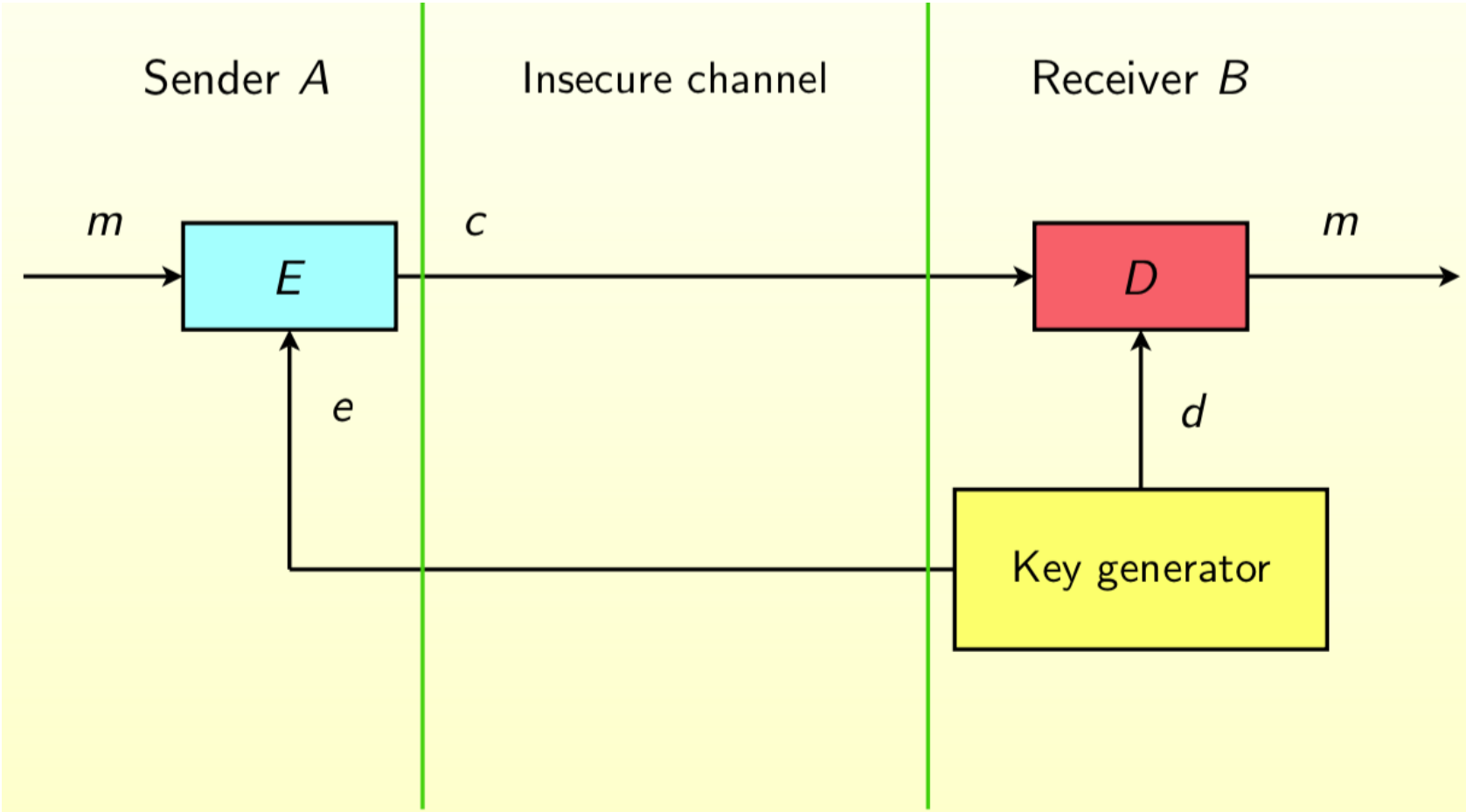
Queensland University of Technology

PART 2 – El Gamal Cryptosystem (1984)



Queensland University of Technology

PART 2 – Rivest-Shamir-Adleman Cryptosystem (1978)



public key $e \in \mathcal{Z}_N$ random

$(\gcd(e, \varphi(N)) = 1)$

$d = e^{-1} \mod (p-1)(q-1)$

Encryption $c = m^e \pmod N$

Decryption $m = c^d \pmod N$

PART 2 – Pairing-based Cryptography

- Pairing invented by Menezes Okamoto and Vanstone (1993) – an attack on elliptic curve logarithms
- Definition:
  Given two abelian groups G_1, G_2 and a cyclic group G_3 of order n, then a pairing is a map

$$e : G_1 \times G_2 \rightarrow G_3$$

  with the following properties:
  - bilinearity

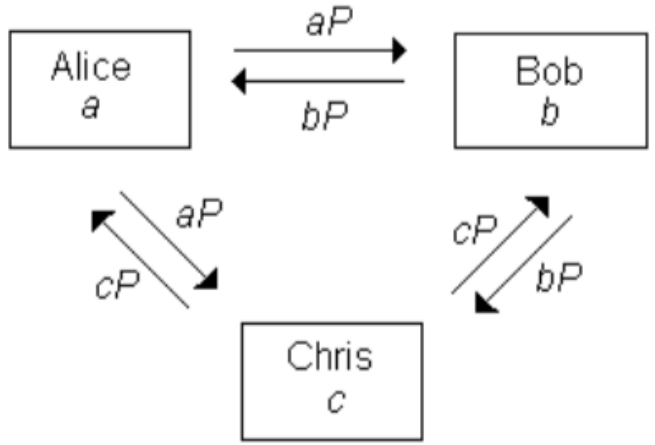
$$\begin{aligned} e(P + P', Q) &= e(P, Q) \cdot e(P', Q) \\ e(P, Q + Q') &= e(P, Q) \cdot e(P, Q') \end{aligned}$$

  - non-degeneracy

$$\begin{aligned} \forall_{P \neq 0; P \in G_1} \exists_{Q \in G_2} \quad & e(P, Q) \neq 1 \\ \forall_{Q \neq 0; Q \in G_2} \exists_{P \in G_1} \quad & e(P, Q) \neq 1 \end{aligned}$$

PART 2 – Three-Party Diffie-Hellman (Joux 2000)

- Alice $\rightarrow$ { Bob, Chris }: $a \cdot P$
- Bob $\rightarrow$ { Alice, Chris }: $b \cdot P$
- Chris $\rightarrow$ { Alice, Bob }: $c \cdot P$



- Alice computes $K = e(b \cdot P, c \cdot P)^a = e(P, P)^{abc}$
- Bob computes $K = e(a \cdot P, c \cdot P)^b = e(P, P)^{abc}$
- Chris computes $K = e(a \cdot P, b \cdot P)^C = e(P, P)^{abc}$

Queensland University of Technology

PART 2 – Identity-Based Encryption (Boneh/Franklin 2001)

- Public-key encryption requires the senders to use AUTHENTIC public keys of receivers
- Need for TA that distributes certificates of public keys (PKI)

Queensland University of Technology

PART 2 – Identity-Based Encryption (Boneh/Franklin 2001)

- IBE
  - Setup:
    $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_3$
    $H_1 : \{0,1\}^* \rightarrow \mathbb{G}_1$ and $H_2 : \{0,1\}^* \rightarrow \mathbb{G}_3$
    Alice's public key $K_A = H_1(ID_A)$
    TA has private key s and public key $S = s * P$
    TA $\rightarrow$ Alice : $D_A = s * K_A$
  - Encryption (by Bob):
    Bob chooses a random r and
    Bob $\rightarrow$ Alice: $R = r * P$ and $c = M \oplus H_2(e(K_A, S)^r)$,
    where hash function G is public
  - Decryption (by Alice):
    $c \oplus H_2(e(D_A, R)) = c \oplus H_2(e(s * K_A, r * P)) = c \oplus H_2(e(K_A, S)^r) = M$

Queensland University of Technology

PART 2 – Identity-Based Encryption (Boneh/Franklin 2001)

- IBE Security:
  - An adversary needs to compute

$$e(K_A, S)^r = e(K_A, P)^{rs}$$

    knowing P, K_A, S and R (hash functions H and G are public)
  - This task is equivalent to solving bilinear Diffie-Helman (BDH) problem

Queensland University of Technology

PART 2 – Certificateless Public-Key Cryptography

- IBE
  - Senders do not need certificates
  - TA generates decryption keys
  - Key escrow problem
  - Revocation could be a problem
- Certificateless (Public-key) Encryption (CE) - Al-Riyami/Paterson 2003
  - Senders do not need certificates
  - No key escrow problem

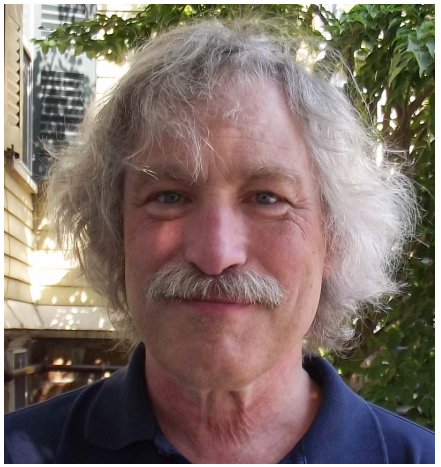
Queensland University of Technology

PART 2 – NTRU Public-Key Encryption

- NTRU – Nth degree TRUncated polynomial ring
- Invented in 1995 by Hoffstein, Pipher, and Silverman
- Let $\mathbb{R} = \mathbb{Z}[x]/(x^n + 1)$ and $\mathbb{R}_q = \mathbb{Z}_q[x]/(x^n + 1)$ be two rings and p, q two primes.
  - Key Generation:
    $f \xleftarrow{\$} D_{Z^n,\sigma}$ s.t. $f = p \cdot f' + 1$
    $g \xleftarrow{\$} D_{Z^n,\sigma}$
    Secret key $sk = f \in \mathbb{R}_q^\times$
    Public key $pk = h = pg/f \in \mathbb{R}_q^\times$
  - Encryption:
    Given message $M = \mathbb{R}/p\mathbb{R}$
    Choose randomly "small" elements $s, e \xleftarrow{\$} \chi_\alpha$
    Cryptogram $C = hs + pe + M \in \mathbb{R}_q$
  - Decryption:
    $C' = f \cdot C$ and $M = C' \mod p$

Queensland University of Technology

Part 2 – Security of NTRU



- The design went through few iterations
- Variant `pNTRUEncrypt` is IND-CPA secure assuming hardness of worse-case problems in ideal lattices (2011, Stehlé and Steinfeld)
- Variant `NTRUCCA` is IND-CCA2 secure assuming hardness of worse-case problems in ideal lattices (2012, Steinfeld et al)
- Invited talk of Jeff Hoffstein at Eurocrypt 2014

Queensland University of Technology

Part 2 – Homomorphic Encryption

- How to secure data in the cloud?
- How to protect privacy if you outsource your computations?
- Homomorphic Encryption - (1978, Rivest, Adleman and Dertouzos)
- How could it work?
  - additive homomorphism

$$E(m_1 + m_2) = E(m_1) + E(m_2)$$

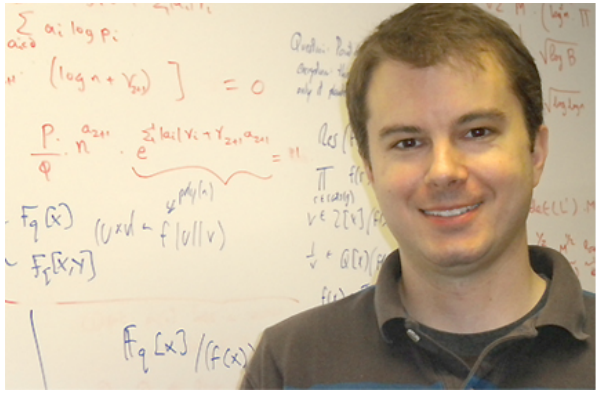
  - multiplicative homomorphisms

$$E(m_1 \cdot m_2) = E(m_1) \cdot E(m_2)$$

- Early homomorphic encryptions:
  - Goldwasser-Micali Encryption (1982)
  - Paillier cryptosystem (1999)

Queensland University of Technology

Part 2 – Fully Homomorphic Encryption



- Fully Homomorphic Encryption (FHE) - Craig Gentry (2009)
- Allows to evaluate a circuit (with addition and multiplication operations) such that

$$f(E(x_1), \ldots, E(x_n)) = E(f(x_1, \ldots, x_n))$$

- Encryption uses lattices and is very slow
- There is 2nd generation of FHE with better efficiency

Queensland University of Technology

Outline

1 Private-key Cryptography

2 Public-key Cryptography

3 Multiparty Computations

4 Cryptanalysis

Queensland University of Technology

Part 3 – Multiparty Computations

Assume that

- there is a collection of participants

$$\{P_1, P_2, \ldots, P_n\} \text{ and a function } Y = F(x_1, \ldots, x_n)$$

- each participant

$$P_i \text{ holds a private input } x_i \text{ for } i = 1, \ldots, n$$

- MPC protocol allows participants to evaluate the function F in such a way that at the end of the protocol
  - all participants learn Y and
  - their inputs remain private

Queensland University of Technology

Part 3 – Ideal Process

Assume that there is a trusted party (TP). Then we can run the following protocol:

- Participants submit their inputs to TP
- TP evaluates the function
- TP distributes the result to all participants

Problem:
What happens if the participants cannot agree on a TP?

Part 3 – Security Settings

Two possible frameworks:

- computationally secure
  breaking the security of the protocol implies that the adversary, in polynomial time, is able to solve a problem that is believed to be intractable
- unconditionally secure
  the adversary cannot break the system by any method better than by guessing private inputs

Part 3 – Adversarial Models

Two generic types

- passive – also called “honest but curious”. The corrupted participants follow the protocol but they try to learn private information
- active – corrupted participants behave arbitrarily and/or maliciously

Queensland University of Technology

Part 3 – Classical Solutions

- Yao, 1982 – the concept of secure MPC – Millionaire Problem
- Goldreich, Micali and Wigderson, 1987 – solution with computational security
- Ben-Or, Goldwasser, and Wigderson and independently Chaum, Crepeau, and Damgård, 1988 – solutions with unconditional security

Part 3 – BGW/CCD Solution

Assume that $Y = F(x_1, \ldots, x_n)$ can be represented by a polynomial (sum of products) over $GP(p)$. The participants

- collectively evaluate products
- collectively evaluate the sums and finding shares of Y

Note 1
At the initial stage, each participant P_i distributes their shares x_i using Shamir secret sharing with the polynomial

$$f_i(x) = x_i + a_1 x + \ldots, + a_t x^t$$

Note 2
Computation of products is highly interactive – the multiplication of two polynomials of degree t gives a polynomial of degree $2t$. Reduction of the degree requires $n \geq 2t + 1$
Note 3

Computation of sums is easy.

Part 3 – Security of MPC

1. In the presence of a passive adversary, no set of size

$$t < n/2$$

of participants learns any additional information, other than what they could derive from their private inputs and the output of the protocol.

2. In the presence of an active adversary, no set of size

$$t < n/3$$

of participants can learn any additional information or disrupt the protocol.

Queensland University of Technology

Part 3 – MPC Applications

- Money without Trusted Authority (Bit Coin)
- Electronic elections
- Collaborative Data Mining
- Lacation-based Services
- Secure Cloud Services
- Electronic Elections

Queensland University of Technology

Outline

1. Private-key Cryptography
2. Public-key Cryptography
3. Multiparty Computations
4. Cryptanalysis

Part 4 – Cryptanalysis

- Linear Cryptanalysis – Mitsuru Matsui 1992 - Story with a “Twist"
- Differential Cryptanalysis – Biham and Shaimr 1989/1990
  Don Coppersmith claims that NSA knew about it in 1974
- Algebraic Cryptanalysis – Courtois and Pieprzyk (2002) limited success for block ciphers – Efficient for stream ciphers
- Cube Attack – Dinur and Shamir 2009 –
  success has many fathers, failure is an orphan –
  Xuejia Lai (High Differentials), Vielbaher (AIDA)

Part 4 – Linear Cryptanalysis (Matsui 1992)

Linear Approximation of S-boxes

Given Boolean function (S-box)

$$f : \{0,1\}^n \rightarrow \{0,1\}^m$$

We can approximate an affine combination of the function outputs $(s_0^*, \ldots, s_{m-1}^*)$ by an affine combination of the inputs $(s_0, \ldots, s_{n-1})$ or more precisely find

$$d(a_0 s_0 \oplus a_1 s_1 \oplus \ldots \oplus a_{n-1} s_{i-1}, \quad b_0 s_0^* \oplus b_1 s_1^* \oplus \ldots \oplus b_{m-1} s_{m-1}^*)$$

where $(a_0, \ldots, a_{n-1})$ and $(b_0, \ldots, b_{m-1})$ are binary strings.

The best linear approximation of S-box is determined by

$$(\tilde{a}_0, \ldots, \tilde{a}_{n-1}) \text{ and } (\tilde{b}_0, \ldots, \tilde{b}_{m-1})$$

such that

$$d(\tilde{a}_0 s_0 \oplus \tilde{a}_1 s_1 \oplus \ldots \oplus \tilde{a}_{n-1} s_{i-1}, \quad \tilde{b}_0 s_0^* \oplus \tilde{b}_1 s_1^* \oplus \ldots \oplus \tilde{b}_{m-1} s_{m-1}^*)$$

is the smallest.

Part 4 – Example - S_5 of DES

If you try all possible input $(s_1, s_2, s_3, s_4, s_5, s_6)$ and output $(s_1^*, s_2^*, s_3^*, s_4^*)$ linear combinations that it turns that there is ONE that is THE BEST. It approximates

$$s_5$$

by

$$s_1^* \oplus s_2^* \oplus s_3^* \oplus s_4^*$$

and

$$d(s_5, \quad s_1^* \oplus s_2^* \oplus s_3^* \oplus s_4^*) = 12$$

We can say that the approximation is TRUE with probability

$$1 - \frac{12}{64} = \frac{52}{64}$$

Queensland University of Technology

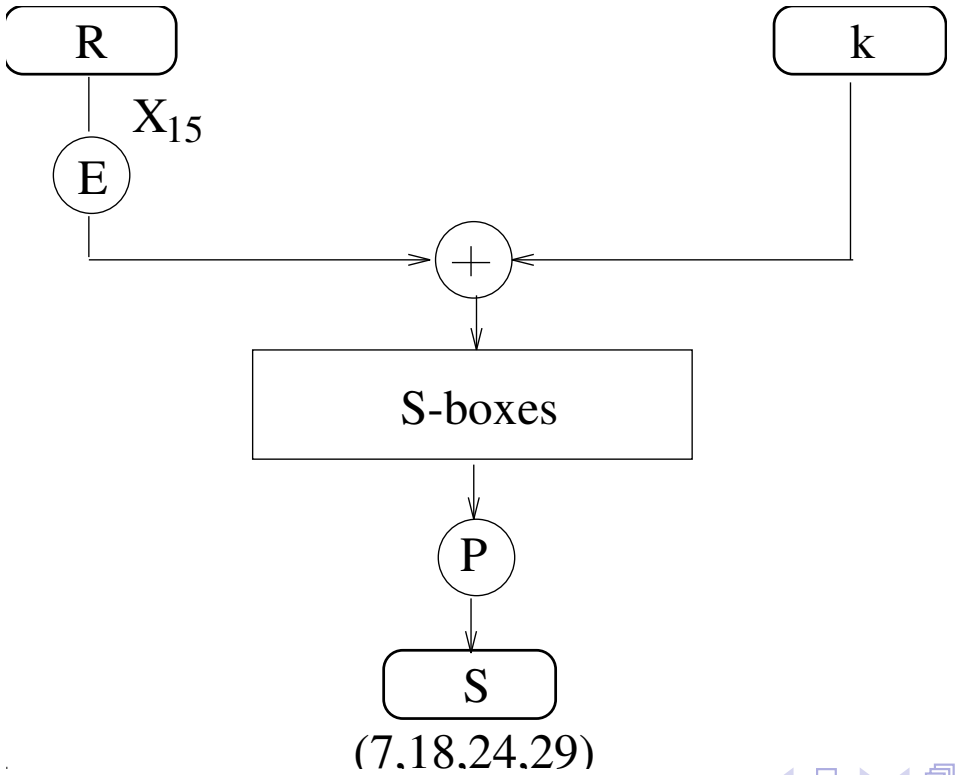
Part 4 – Linear Cryptanalysis of 3-Round DES

We use the best linear approximation for S_5

$$s_5 = s_4^* \oplus s_3^* \oplus s_2^* \oplus s_1^* \text{ with the probability } \frac{52}{64}$$

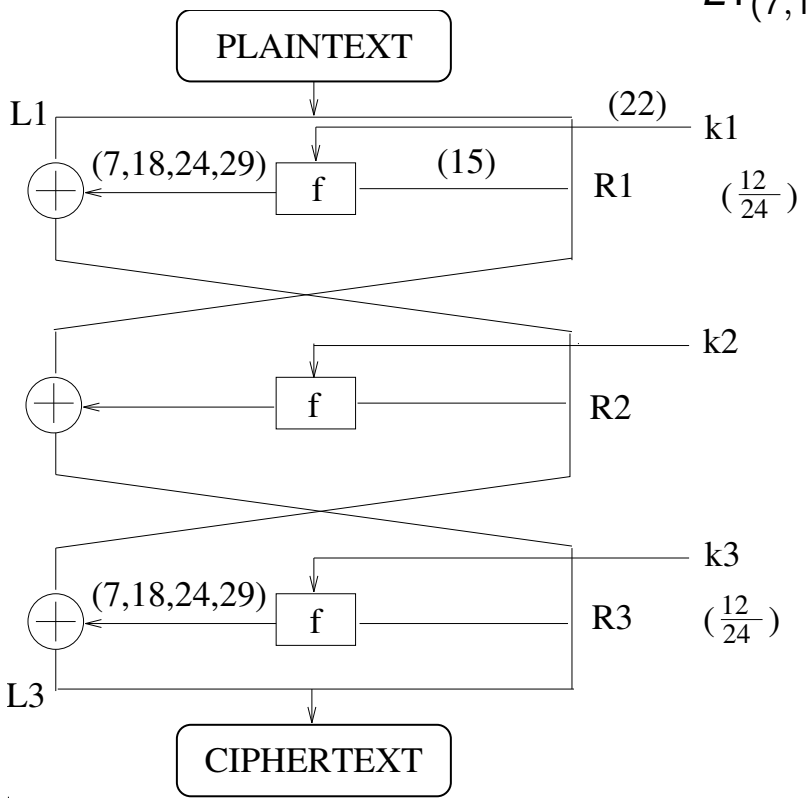
The linear approximation gives the following relation

$$R_{(15)} \oplus k_{(22)} = S_{(7)} \oplus S_{(18)} \oplus S_{(24)} \oplus S_{(29)} \stackrel{def}{=} S_{(7,18,24,29)}$$



Queensland University of Technology

Part 4 – Linear Cryptanalysis of 3-Round DES



$$L1_{(7,18,24,29)} \oplus R2_{(7,18,24,29)} = k1_{(22)} \oplus R1_{(15)} \rightarrow \frac{52}{64}$$

$$L3_{(7,18,24,29)} \oplus R2_{(7,18,24,29)} = k3_{(22)} \oplus R3_{(15)} \rightarrow \frac{52}{64}$$

How we can combine the two (probabilistic) relations so the internal variables $R2_{(7,18,24,29)}$ gets cancelled?

Queensland University of Technology

Part 4 – Linear Cryptanalysis of 3-Round DES

Matsui Piling-up Lemma

Theorem

Given two binary random variables X_1 and X_2 whose probabilities are $P(X_1 = 0) = p$ and $P(X_2 = 0) = q$, then

$$P(X_1 \oplus X_2 = 0) = pq + (1 - p)(1 - q)$$

Proof:
The random variable $X_1 \oplus X_2 = 0$ iff

- both $X_1 = 0$ and $X_2 = 0 \rightarrow$ this happens with the probability pq
- both $X_1 = 1$ and $X_2 = 1 \rightarrow$ this happens with the probability $(1 - p)(1 - q)$

Queensland University of Technology

Part 4 – Linear Cryptanalysis of 3-Round DES

So the approximations

$$L1_{(7,18,24,29)} \oplus R2_{(7,18,24,29)} = k1_{(22)} \oplus R1_{(15)} \rightarrow \frac{52}{64}$$

$$L3_{(7,18,24,29)} \oplus R2_{(7,18,24,29)} = k3_{(22)} \oplus R3_{(15)} \rightarrow \frac{52}{64}$$

can be combined giving the approximation also called linear characteristic

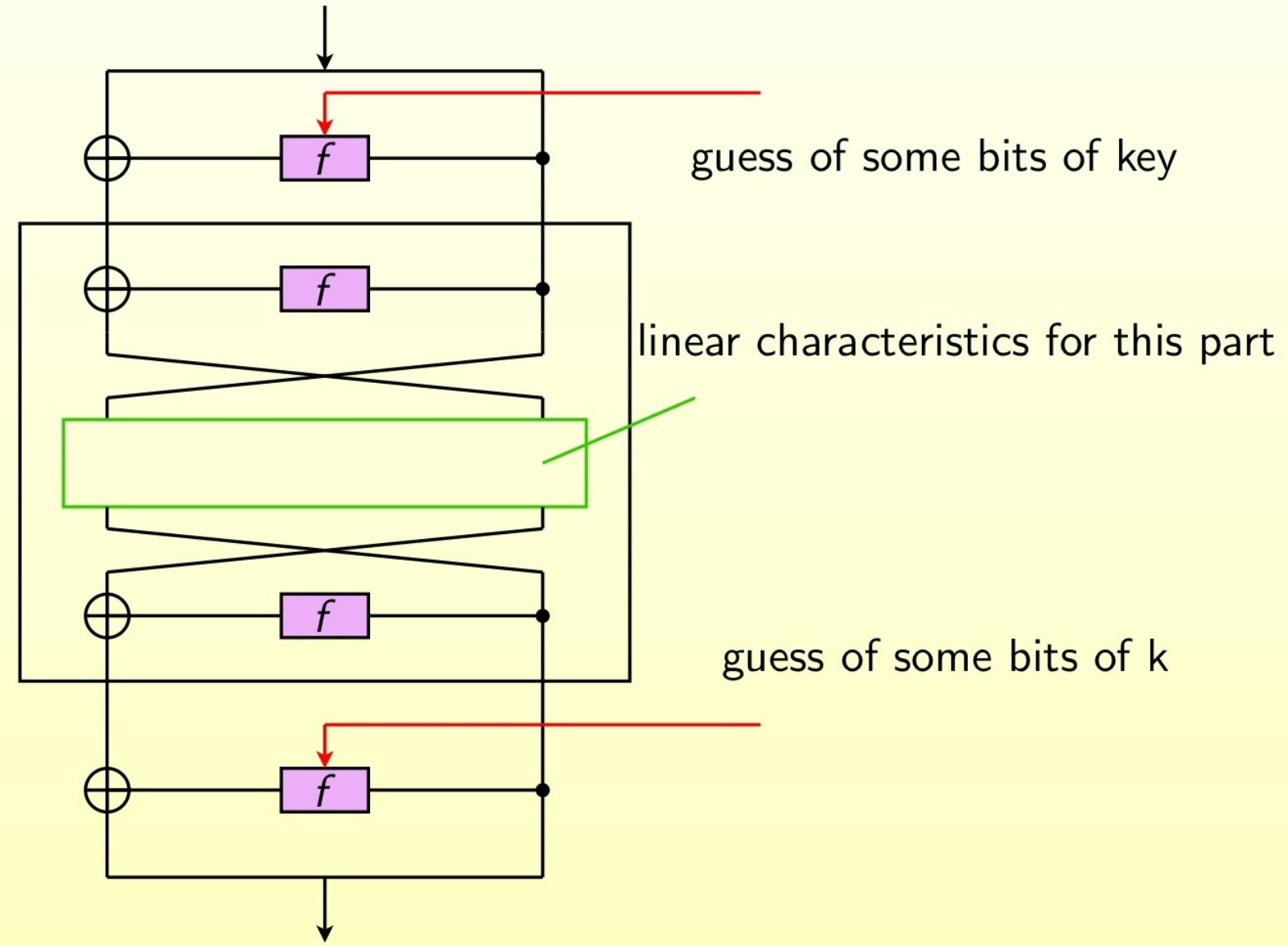
$$L1_{(7,18,24,29)} \oplus L3_{(7,18,24,29)} = k1_{(22)} \oplus R1_{(15)} \oplus k3_{(22)} \oplus R3_{(15)}$$

that is true with the probability

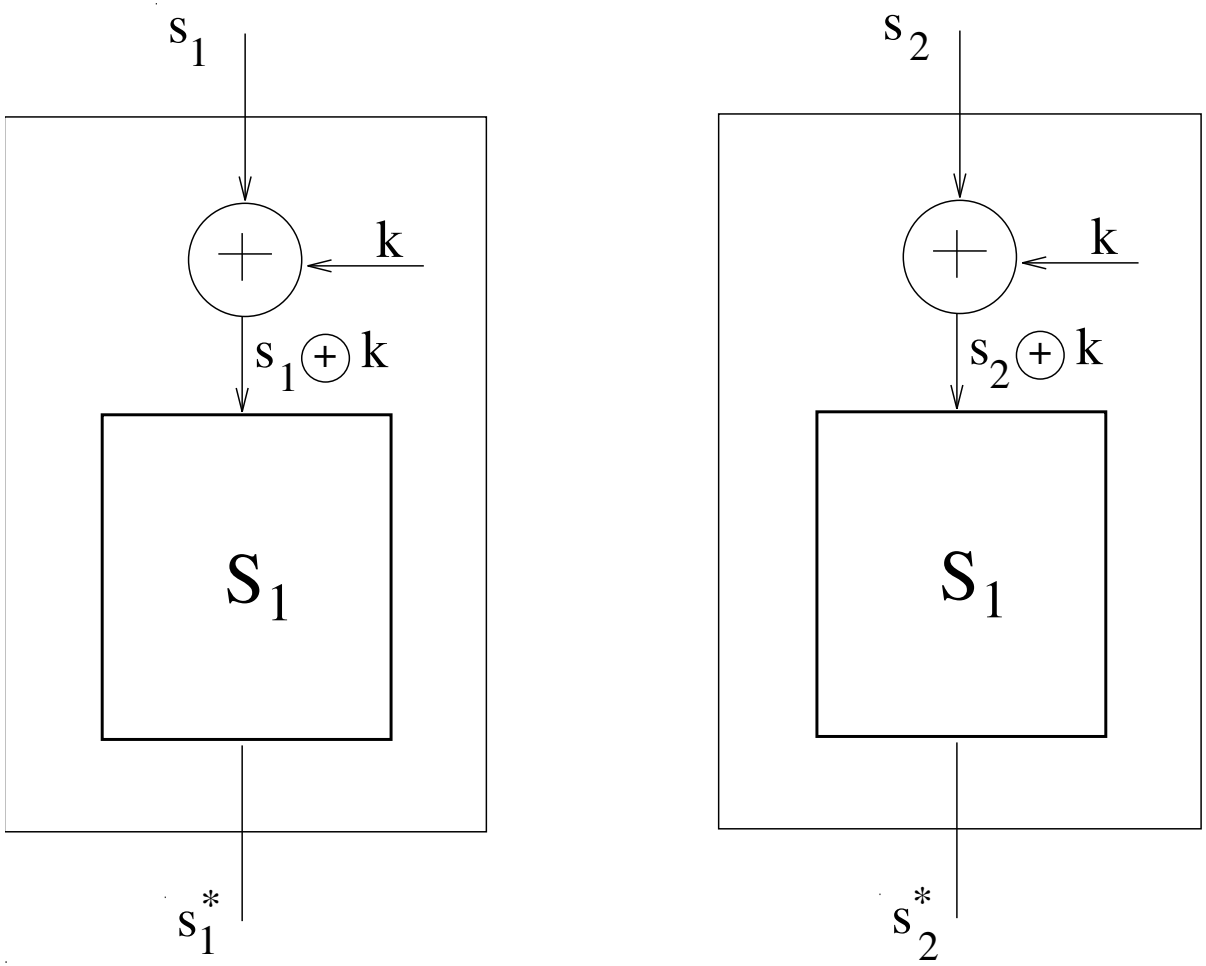
$$\left(\frac{52}{64}\right)^2 + \left(\frac{12}{64}\right)^2 \approx 0.7$$

Queensland University of Technology

Part 4 – Linear Cryptanalysis – Matsui's Trick



Part 4 – Differential Cryptanalysis (Biham and Shamir, 1990)



Queensland University of Technology

Part 4 – Differential Cryptanalysis – XOR Profile

Given S-box as follows:

Input	Output
s	s^*
000	000
001	010
010	001
011	100
100	101
101	011
110	111
111	110

Queensland University of Technology

Part 4 – Differential Cryptanalysis – XOR Profile

Let us find entries for the row with $\delta = 001$

s_1	$s_2 = s_1 \oplus \delta$	$s_1^* = S(s_1)$	$s_2^* = S(s_2)$	Δ
000	001	000	010	010
001	000	010	000	010
010	011	001	100	101
011	010	100	001	101
100	101	101	011	110
101	100	011	101	110
110	111	111	110	001
111	110	110	111	001

$\delta \setminus \Delta$	000	001	010	011	100	101	110	111
000	8	0	0	0	0	0	0	0
001	0	2	2	0	0	2	2	0
⋮								

Queensland University of Technology

Part 4 – Differential Cryptanalysis – XOR Profile

IMPORTANT OBSERVATION – the XOR profiles of

- $S(x)$ and
- $S(x \oplus k)$

are identical ("modulo a permutation of columns")

Assumptions about the attacker - she knows

- S-box (its truth table)
- input values (messages)
- output differences

Queensland University of Technology

Part 4 – Differential Cryptanalysis – Simple Attack

Given S-box as follows:

Input s	Output s^*
000	000
001	010
010	001
011	100
100	101
101	011
110	111
111	110

Adversary knows

- pair of observed inputs ($s_1 = 101$, $s_2 = 110$)
- output difference observed $\Delta = s_1^* \oplus s_2^* = 100$

Note that $\delta = s_1 \oplus s_2 = 011$.
What does Adversary learn about the secret key?

Queensland University of Technology

Part 1 – Differential Cryptanalysis – Simple Attack

Finding the set $\mathcal{S}_{\Delta}^{\delta} = \mathcal{S}_{100}^{011}$

$s_1 \oplus k$	s_1^*	$s_2 \oplus k = s_1 \oplus k \oplus \delta$	$s_2^* = S(s_2 \oplus k)$	Δ
000	000	011	100	**100**
001	010	010	001	011
010	001	001	010	011
011	100	000	000	**100**
100	101	111	110	011
101	011	110	111	**100**
110	111	101	011	**100**
111	110	100	101	011

$\mathcal{S}_{100}^{011} = \{000, 011, 101, 110\}$

Note that

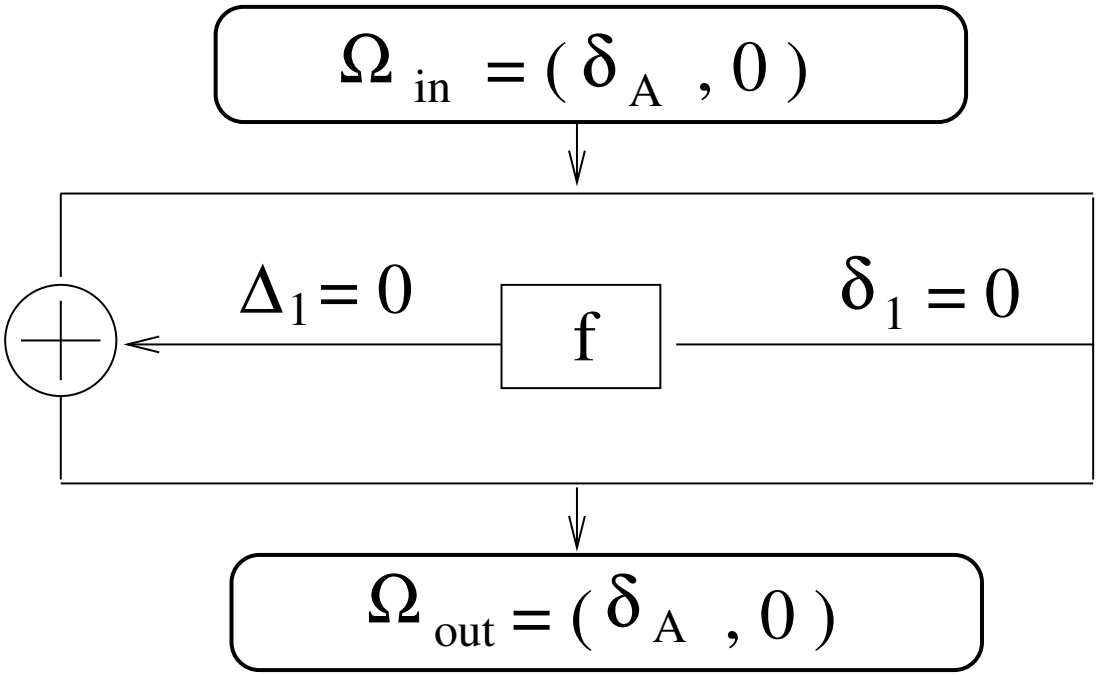
$$k \in \mathcal{S}_{100}^{011} \oplus s_1 = \{101, 110, 000, 011\}$$

and

$$k \in \mathcal{S}_{100}^{011} \oplus s_2 = \{110, 101, 011, 000\}$$

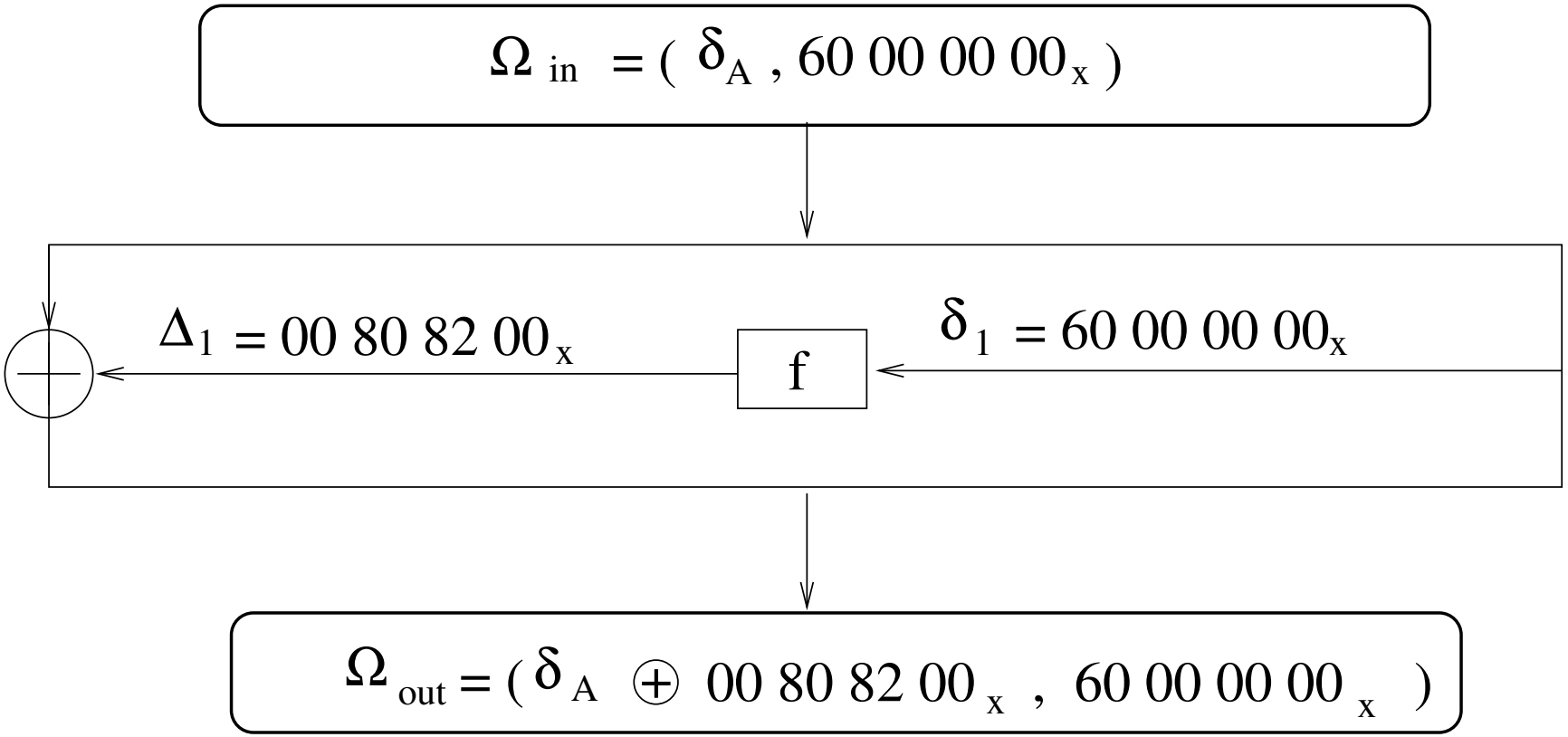
Queensland University of Technology

Part 4 – Differential Cryptanalysis – Single Round Characteristics



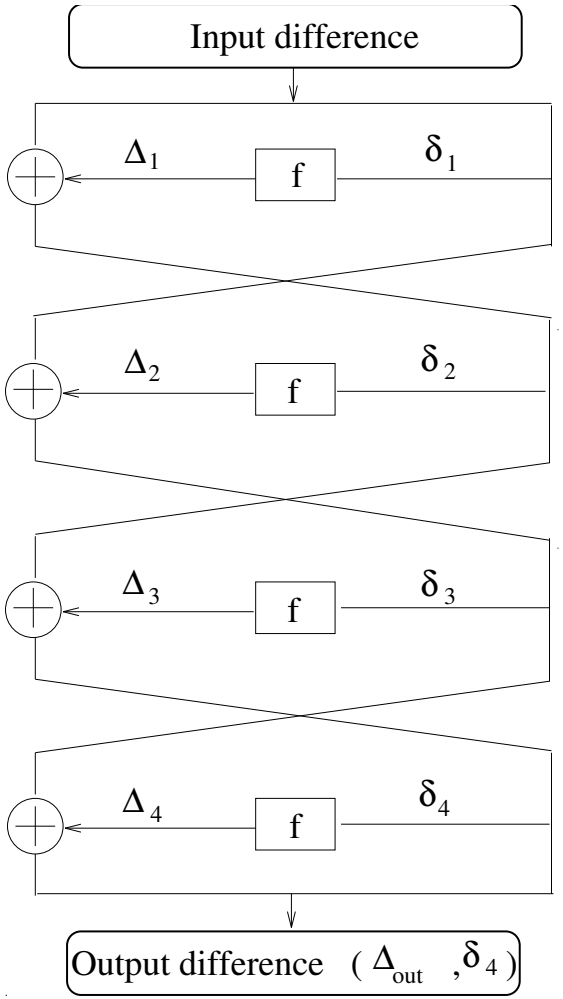
Queensland University of Technology

Part 1 – Differential Cryptanalysis – Other Single Round Characteristics



Queensland University of Technology

Part 4 – Differential Cryptanalysis of 4-Round DES



Assume that we have a pair of messages at random with their input differences

$$(\delta_A = 20\ 00\ 00\ 00 \text{ and } \delta_1 = 00\ 00\ 00\ 00)$$

Note that the output left-hand difference

$$\Delta_{out} = \Delta_4 \oplus \Delta_2 \oplus \delta_1$$

As $\delta_1 = 0$, then

$$\Delta_4 = \Delta_{out} \oplus \Delta_2$$

As we know 28 bits of Δ_2 and observe $\Delta_{out} \Rightarrow$ we know 28 bits of Δ_4.
By differential cryptanalysis, we can recover $6 \times 7 = 42$ bits of working key k_4 used in 4-th round.

Queensland University of Technology

Conclusions

- Cryptography is as good as its implementation (side-channel attacks)
- Human factor always crucial (education)
- Dual-use technologies - controversy in Australia
- Use of strong cryptography as protection of citizens' privacy – panel discussion at Eurocrypt 2014

Queensland University of Technology