Generic Attacks on Symmetric Ciphers

Palash Sarkar

Indian Statistical Institute

email: palash@isical.ac.in

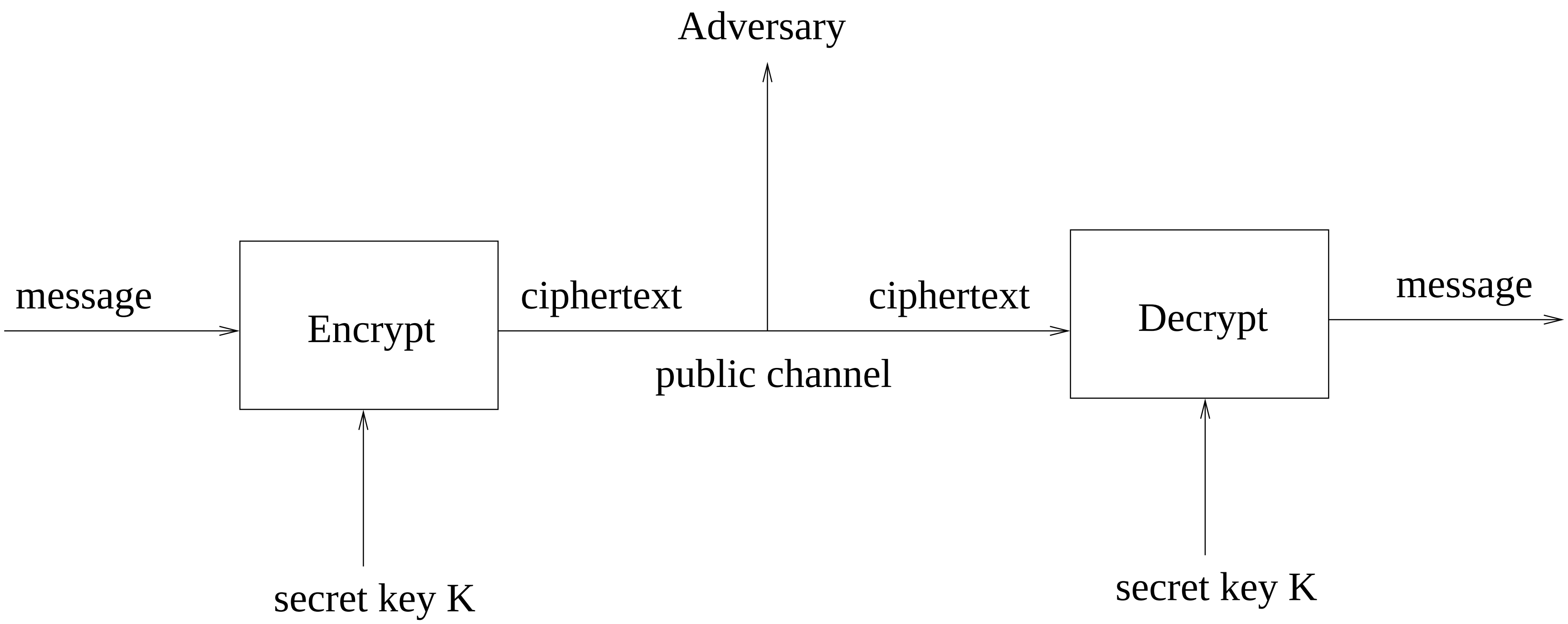
Indocrypt

The 7th International Conference on Cryptology in India

Kolkata, India

10th December, 2006.

Symmetric Key Encryption



Adversary's goal: To obtain secret key K.

Generic Attacks

- Most basic of all attacks.
- Treats the algorithm as a black box.
- The same attack idea can be used for many algorithms.
- One can utilise a large amount of parallelism.
- Most feasible implementation in special purpose hardware.
- There are some reported implementations.

Non-Generic Attacks

- Exploits special properties of an algorithm or a class of algorithms.
- Examples: Linear and differential cryptanalysis, correlation attacks, algebraic attacks, etcetra.
- The literature abounds in such attacks and ideas.
- Very little (or no) work on
  - hardware implementation of non-generic attacks;
  - exploiting parallelism.

Presentation Outline

- Exhaustive search attacks.
  - Exhaustive search on DES.
  - DES Cracker by Electronics Frontier Foundation (EFF).
- Time/Memory trade-off (TMTO) attacks.
  - Hellman's algorithm.
  - Rivest's distinguished point method.
  - Biryukov-Shamir multiple data method.
  - Other variants.

Presentation Outline (contd.)

- A Cost Analysis of TMTO Attack.
  - Strength of different key sizes.
- Non-Conventional Techniques for Exhaustive Search.
  - Based on DNA computing.
  - Based on photographic techniques.
- Rejewski's Attack on Enigma.
  - Unearthing some modern ideas in the old attack.

Exhaustive Search

Known Plaintext:

- A pair (M, C) is available.
- Attack: Decrypt C by the possible keys one-by-one until M is obtained.

Ciphertext Only:

- Only C is available.
- Attack: Decrypt C by the possible keys one-by-one until a “meaningful” message is obtained.

Parallelism:

- Decryptions by separate keys are unrelated.
- Efficiently parallelizable, subject to resource restrictions.

Exhaustive Search on DES

Diffie-Hellman (1977):

Special purpose hardware;
Estimated time 12-hours at a cost of USD 20M.

Wiener (1993):

Special purpose hardware;
Estimated time 3.5 hours at a cost of USD $1M$.

Goldberg-Wagner (1996):

FPGA based hardware;
Estimated time one year at a cost of USD $45,000$.

Exhaustive Search on DES

Response to challenges announced at RSA Cryptographic Trade Shows.

1997:

Software – using computers distributed on the internet;
Time required five months.

1998:

Software – using computers distributed on the internet;
Time required 39 days.

Exhaustive Attacks on DES

Electronic Frontier Foundation (EFF) (July 17, 1998): DES Cracker
Special purpose hardware;
Time required: Solved "DES Challenge II" in less than 3 days.
Cost: USD $200,000$ (hardware + development)

Quisquater-Standaert (2005):
Time/Memory Trade-Off;
Estimated cost of USD $12,000$ and online time half an hour.

Contrast: 100,000 British pounds to build one cryptanalytic machines to attack Engima ($\approx$ 1941). ("The Code Book" by Simon Singh, Page 174).

DES Cracker – Basic Design

- DES key space size = 2^{56}.
- Ciphertext only attack.
- Hierarchical design.
  - Parallel search units controlled by a single computer.
  - 24 search units fit inside a chip.
  - 64 chips are mounted on a large board.
  - 12 boards are mounted onto a chassis.
  - 2 chassis are used.
- Total of 36,000 search units.

One Search Unit

- Includes a 32-bit counter to generate candidate keys.
- Top 24 bits loaded by the computer.
  - Total key size is 56 bits.
  - A direct implementation of a 56-bit counter was considered to be too costly.
- Interrupts the computer after searching 2^{32} keys.
- Reports “interesting” plaintext to the computer.
- Final decision taken by the computer.

Suggested Improvement

- Use of 56-bit LFSR to generate candidate keys.
- Easy to partition total key space into disjoint sets.
- Parallel generation of the subsets is easy.
- Each search unit has a separate implementation of the LFSR.
- Advantages:
  - Next key in one clock cycle.
  - Avoids the 2^{24} interrupts made by the search units to the computer.
  - Leads to a simpler, smaller and faster design.

Recent: Mukhopadhyay-Sarkar, 2006.
Earlier: Wiener, 1993; Goldberg-Wagner 1996.

Time/Memory Trade-Off

- Introduced by Hellman in 1980.
- Basic idea:
  - Perform an exhaustive search (or a little more computation) once.
  - Store some of the results of intermediate computations.
  - At a later stage use the stored results to obtain secrets *significantly* faster than exhaustive search.
- In its general form, a TMTO inverts a one-way function.

Basic Hellman Method

- Define $f(K) = E_K(M)$ for a fixed M. Inverting $f()$ on $C = E_K(M)$ yields K.
- **Pre-Computation:**
  - Compute r tables of size $m \times t$ each.
  - For each table, store only the first and the last column, sorted on the last column.
- **Online Search:**
  - Given y, we have to find x such that $f(x) = y$.
  - Perform some computations and look-ups on the stored tables to find x.
- Success depends on proper choice of the parameters r, t and m.

Single Table Computation

The domain and range of f are same and of size N.

- **Chain:** Given x, the chain starting at x is

$$x, f(x), f^2(x), f^3(x), \ldots, f^t(x)$$

- Choose m points $x_{1,0}, \ldots, x_{m,0}$ randomly.
- Construct chains (rows of the table) starting at these points.
- The start points and the end points are stored, sorted on the end points.

A Hellman Table

$$
\begin{array}{|c|ccccccccccc|c|}
\cline{1-1}\cline{13-13}
x_{1,0} & \xrightarrow{f} & x_{1,1} & \xrightarrow{f} & x_{1,2} & \xrightarrow{f} & \cdots\cdots & \xrightarrow{f} & x_{1,t-1} & \xrightarrow{f} & & & x_{1,t} \\
x_{2,0} & \xrightarrow{f} & x_{2,1} & \xrightarrow{f} & x_{2,2} & \xrightarrow{f} & \cdots\cdots & \xrightarrow{f} & x_{2,t-1} & \xrightarrow{f} & & & x_{2,t} \\
x_{3,0} & \xrightarrow{f} & x_{3,1} & \xrightarrow{f} & x_{3,2} & \xrightarrow{f} & \cdots\cdots & \xrightarrow{f} & x_{3,t-1} & \xrightarrow{f} & & & x_{3,t} \\
\vdots & & \vdots & & \vdots & & & & \vdots & & & & \vdots \\
x_{m,0} & \xrightarrow{f} & x_{m,1} & \xrightarrow{f} & x_{m,2} & \xrightarrow{f} & \cdots\cdots & \xrightarrow{f} & x_{m,t-1} & \xrightarrow{f} & & & x_{m,t} \\
\cline{1-1}\cline{13-13}
\end{array}
$$

Start Points

End Points

Set of Tables

- One table cannot have too many rows, as then repetition will occur.
- Let $\phi_1, \ldots, \phi_r$ be very simple functions.
- Define $f_i = \phi_i \circ f$.
- We *assume* that the f_i's behave as independent random functions.
- Construct r tables using the functions $f_1, \ldots, f_r$.
- We wish to have $rmt = N$.

Online Search

Search for y in the ith table. Repeat for each table.

- Let $z = \phi_i(y)$.
- If z is in the ith table, then its predecessor in the chain (row) is a pre-image of y.
- Compute

$$z_0 = z, z_1 = f_i(z), z_2 = f_i^2(z), \ldots, z_t = f_i^t(z)$$

- Look-up each z_j among the end-points of the ith table.
- If found, then from the corresponding start-point, generate the chain upto z_0.
- The predecessor of z_0 is a pre-image of y.

Coverage and Analysis

- Constants are ignored.
- $mt^2 \leq N$ (birthday paradox).
- $rmt = N$ (total coverage).
- **Pre-Computation:** Number of f invocations = N (exhaustive search).
- **Memory:** $M = 2rm$.
- **Online search:** $T = rt$;
  Max number of f invocations = rt;
  Max number of table look-ups = rt.
- $MT^2 = N^2$ (the trade-off curve).
  $(M = T = N^{2/3}.)$

Distinguished Points

Introduced by Rivest.

Distinguished point (DP): A string with a fixed number of bits 0.

Pre-Computation: Generate a chain only upto a DP, or of length t whichever occurs earlier.

Online Search: Perform a look-up only after encountering a DP.

- Only one look-up for each table required.
- Total number of look-up comes down to r.

Chains are of variable length.

Trade-off curve does not change.

TMTO With Multiple Data

Biryukov-Shamir, 2000.

- **Given**: $y_1, \ldots, y_D$.
  Requirement: Invert any of the y_i's.
- Choose $r = t/D$.
- Coverage $rmt = N/D$. Birthday paradox assures a constant success probability.
- Trade-off curve: $TM^2D^2 = N^2$.

Biryukov-Mukhopadhyay-Sarkar (2005) presents a detailed analysis.

Rainbow Method

Oechslin, 2003.

- It is a variant of the Hellman method.
- **Rainbow chain:** $x_0 = x$.

  $$x_1 = f_1(x_0), x_2 = f_2(x_1), \ldots, x_t = f_t(x_{t-1}).$$

  Recall: $f_i()$ is obtained from $f()$ by a simple output modification.
- A single rainbow table of size $mt \times t$ is used.
- Online search time is one-half of Hellman's method.
- Trade-off curve: $TM^2D = N^2$.
  Biryukov-Mukhopadhyay-Sarkar (2005).

Rainbow Crack

- A software implementation of the rainbow method.
- Pre-computed tables available on the net.
- Using these, the program can crack Windows passwords in a few seconds of online time.
- A vindication of the TMTO method.

The following website provides a real-life demo.

http://lasecwww.epfl.ch/˜oechslin/projects/ophcrack/

Theoretical Issues

- Hellman's assumption:
  - f is a random function.
  - $f_1, \ldots, f_t$ are independently chosen random functions.
- Fiat-Naor (1991) counter-example f:
  - $N - N^{1-\epsilon}$ points induce a permutation.
  - Other keys map to zero.
- Hellman's method on such f results in a lot of zeros in the tables.
- Success probability drops.

Inverting Any Function

Fiat-Naor 1991.

- Store high in-degree images in a single table.
- For the other points use a variant of Hellman's strategy.
- Can provably invert any function.
- Trade-off Curve: $TM^3 = N^3$.
- Worse trade-off compared to Hellman's method.

An Easy Solution

- Output modifications.
  - Hellman: Permute output bits.
  - Oechslin: $f_i(x) = f(x) \oplus i$.
  - For both methods, it is possible to construct Fiat-Naor type examples.
- A new output modifier (Mukhopadhyay-Sarkar 2005).
  - Randomly choose a maximal length LFSR.
  - Suppose LFSR states are $X_1, X_2, \ldots$.
  - Define $f_i(x) = f(x) \oplus X_i$.
  - Difficult to construct a Fiat-Naor type example for this construction.

Applying TMTO

Identifying one-way functions in cryptographic algorithms.

- Hellman: Block ciphers. For a fixed M

$$f(K) = E_K(M).$$

- Babbage, Golić, Biryukov-Shamir: Stream ciphers.

$$\text{internal state} \mapsto \text{keystream-segment map.}$$

- Biryukov-Shamir-Wagner: A5/1 stream cipher.
  - A variant of distinguished point method.
  - Introduces a new technique called BSW sampling.

Applying TMTO

- Hong-Sarkar:
  - Stream ciphers.

$$\text{(Key, IV)} \mapsto \text{keystream-segment map.}$$

  - Several modes of operations of block ciphers.
  - A general method of applying TMTO.
- Biryukov-Mukhopadhyay-Sarkar:
  - Unix password.
- Other applications are known.

Parallelism

- DES Cracker uses a large amount of parallelism.
- Hardware designs for TMTO are parallel.
- Bernstein 2005.
  - Points out that a clever but *sequential* attack can be inferior to brute force but *parallel* search.
  - Writes Hellman+DP and rainbow methods as exhaustive search attacks.

Processors and Bounds

- Amirazizi-Hellman 1988: Introduced time/memory/processor trade-offs.
  - Uses a number of processors.
  - Uses a large shared memory.
  - Uses a switching/sorting network.
- Wiener 2004: Notion of full cost (Lenstra et al 2002, Bernstein 2001).

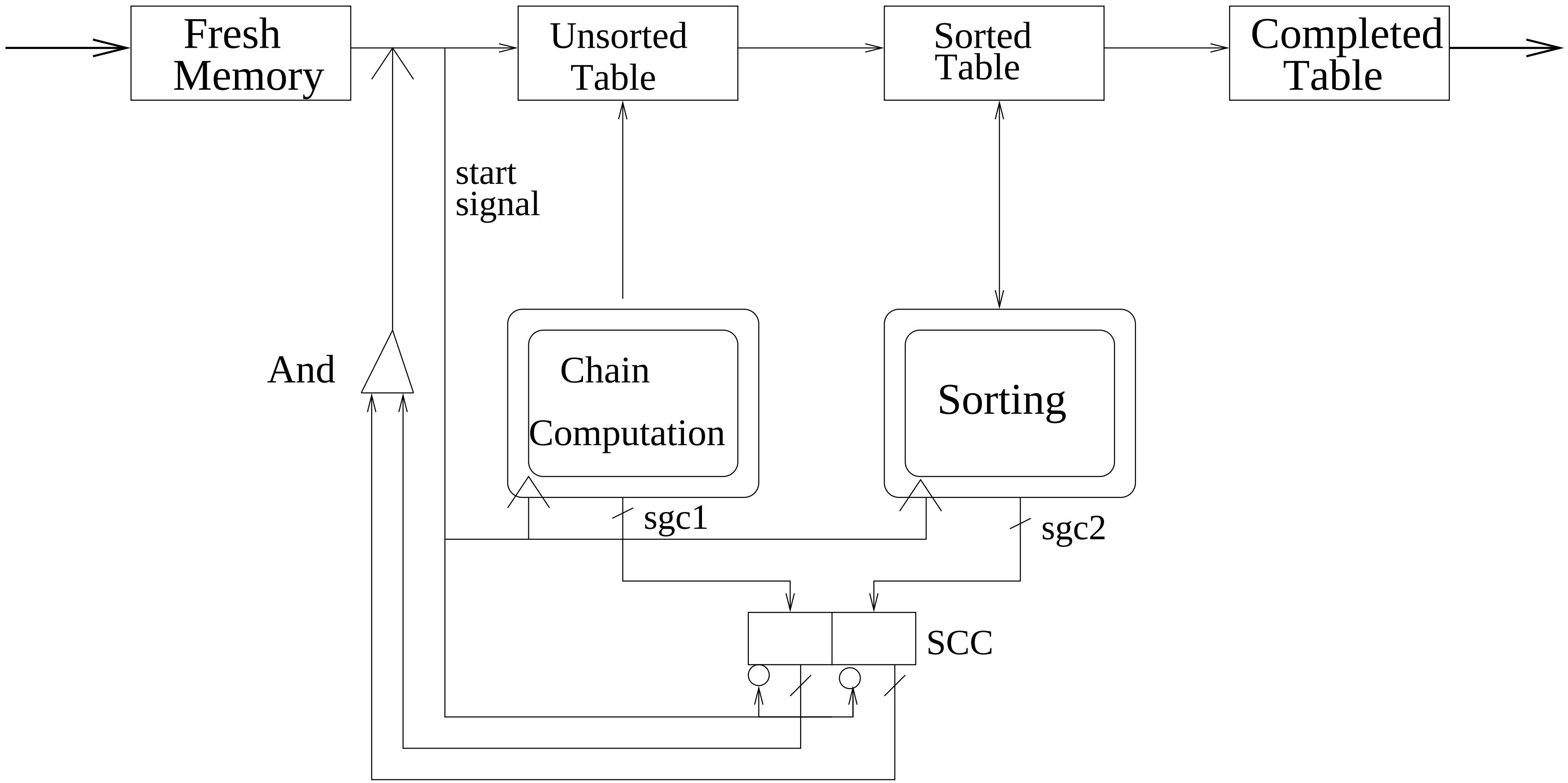
$$\text{number of components} \times \text{duration of their use}$$

  - A bound of $\theta(n^{3/2})$ on the interconnection cost for connecting n processors to n memory blocks.

TMTO Architecture

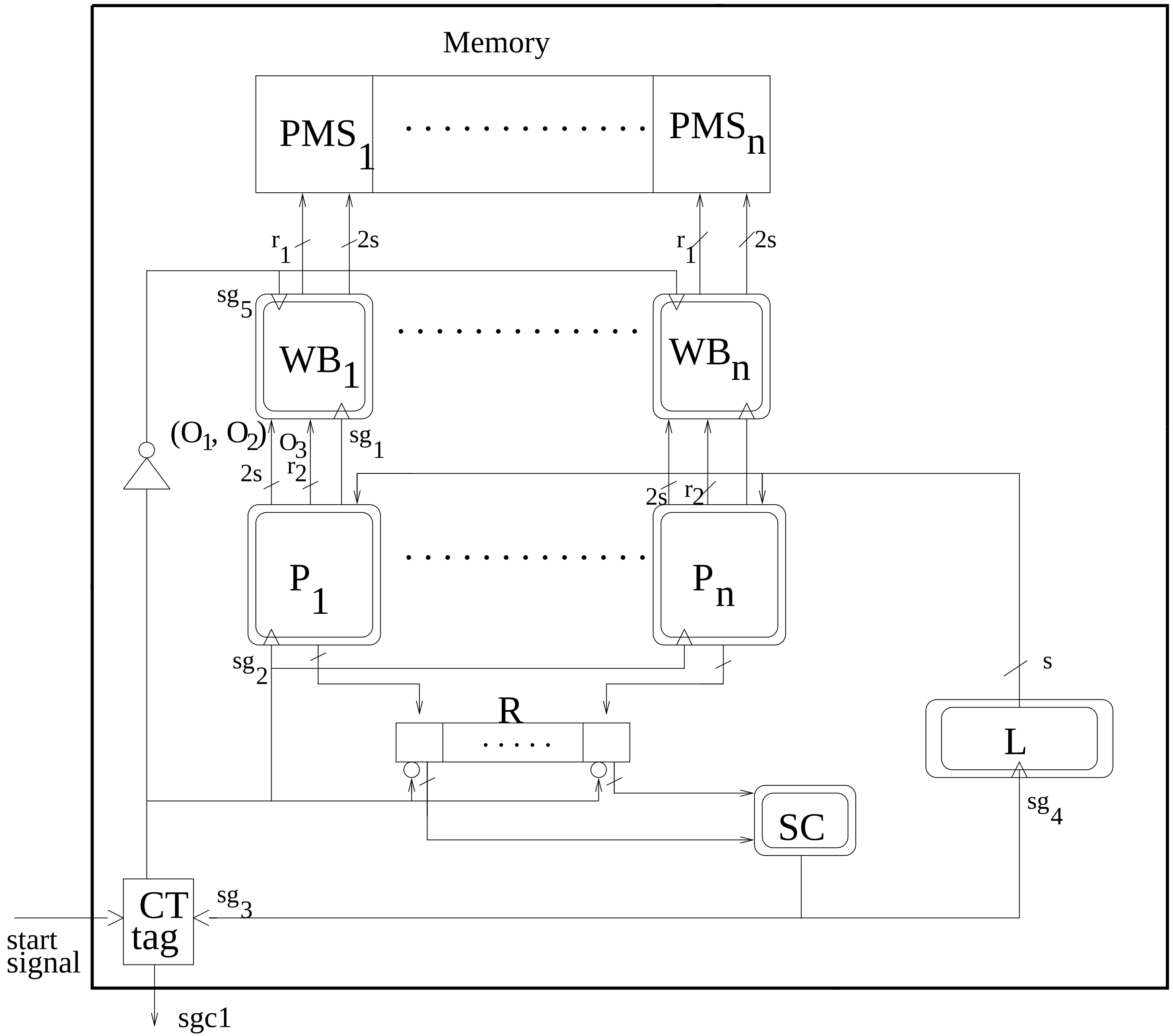
- FPGA implementations:
  - Standaert et al (2002): 40-bit DES keys.
  - Mentens et al (2006): 48-bit UNIX passwords.
  - Use of counters for start point generation.
- Special purpose hardware: (Top-level design.) Mukhopadhyay-Sarkar 2006.
  - A different architecture for avoiding lower bound on interconnection cost.
  - Use of LFSRs for start point generation and function masking.
  - Store one table on a DVD.

Pre-Computation: Basic Blocks

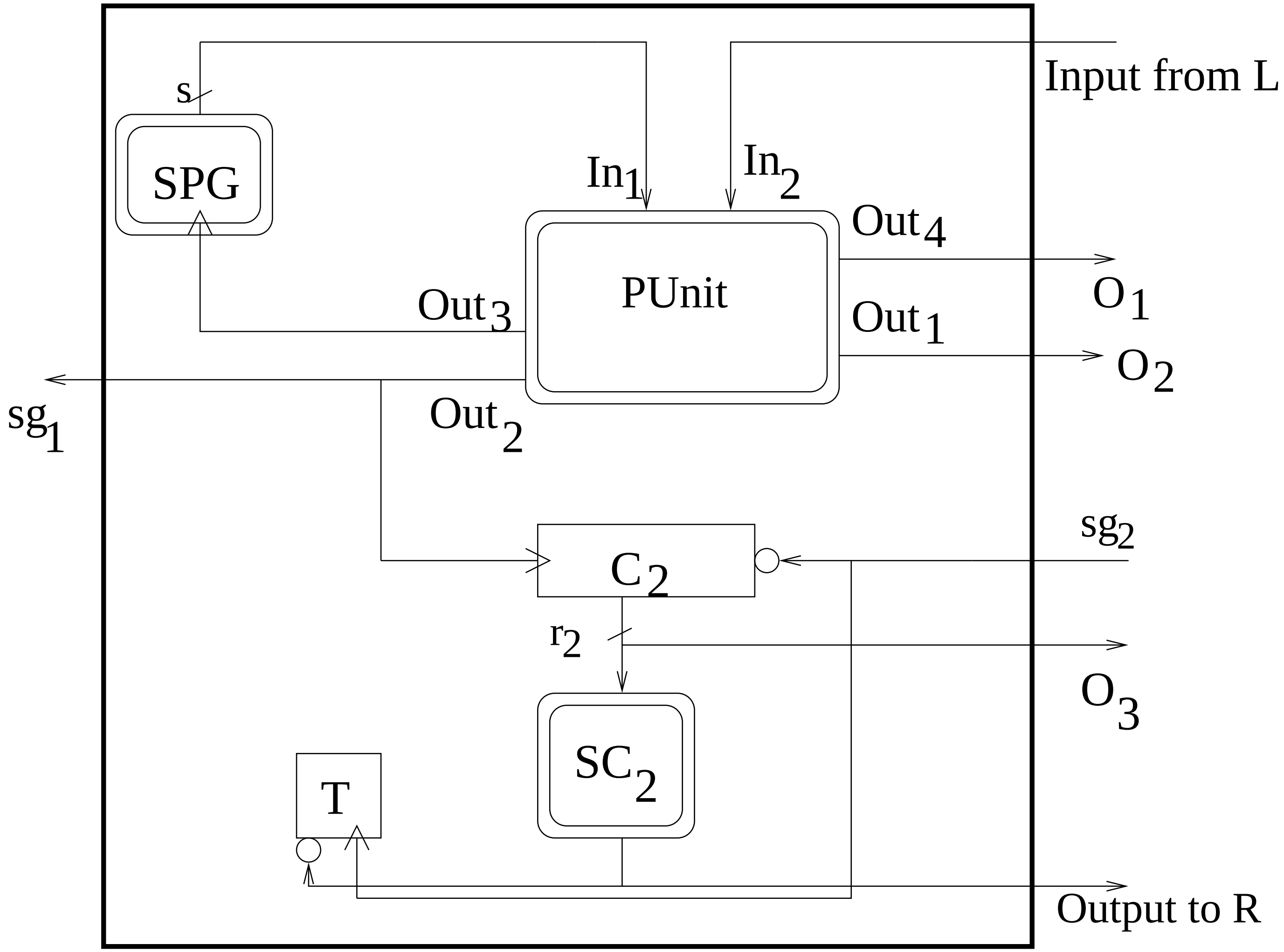


Assembly line processing.

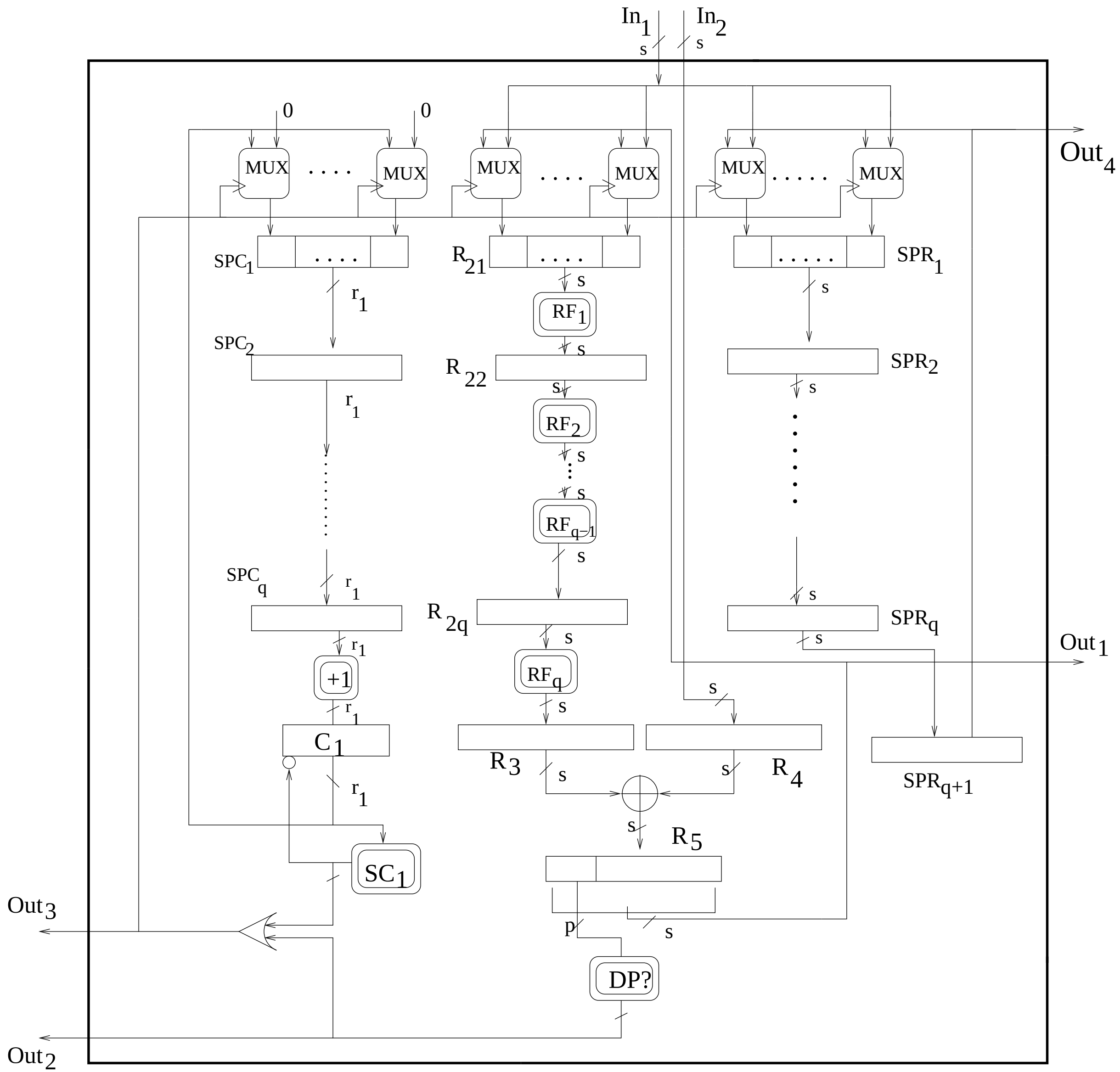
Chain Computation



Architecture of each P_i

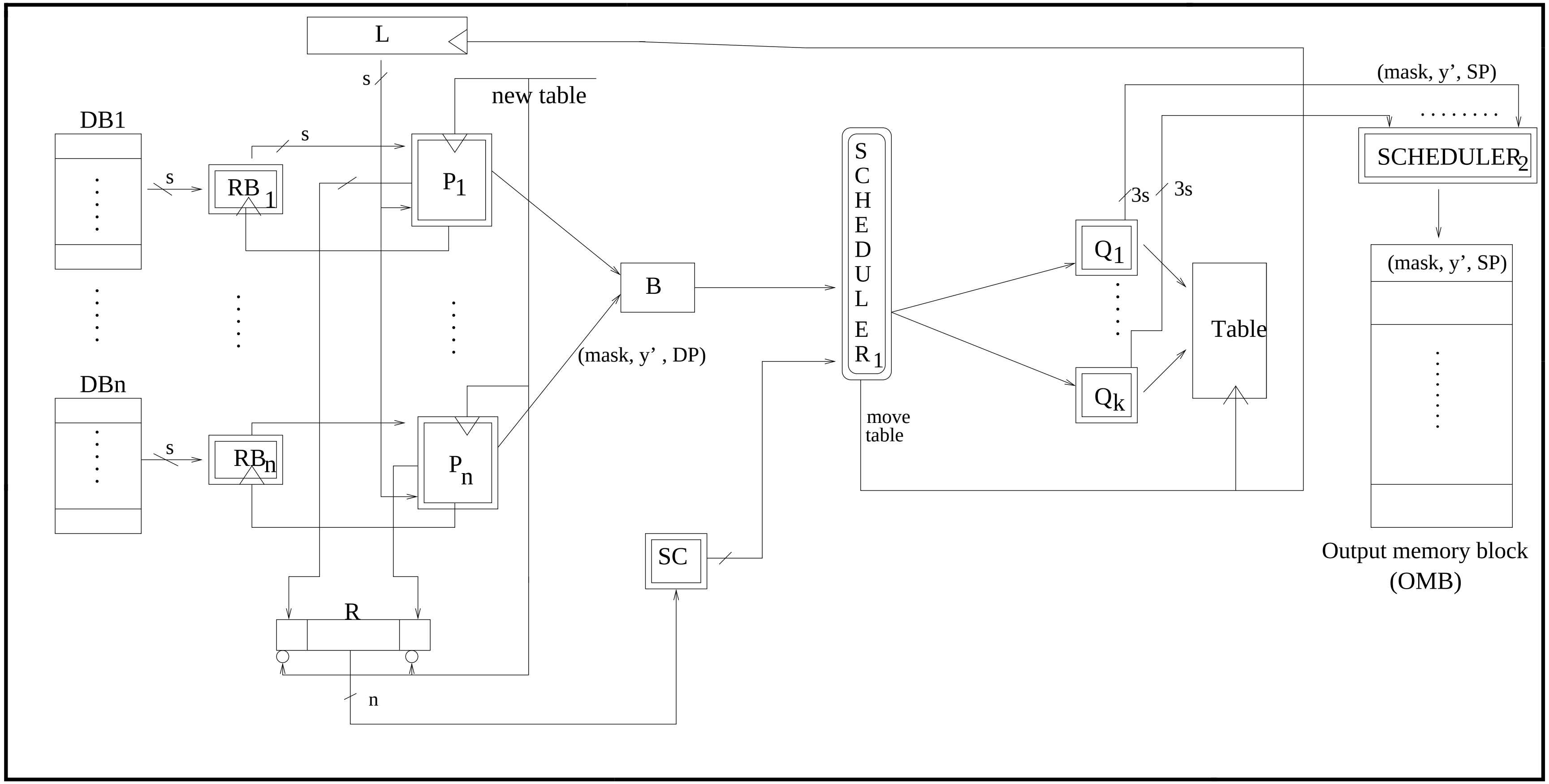


Architecture of Punit



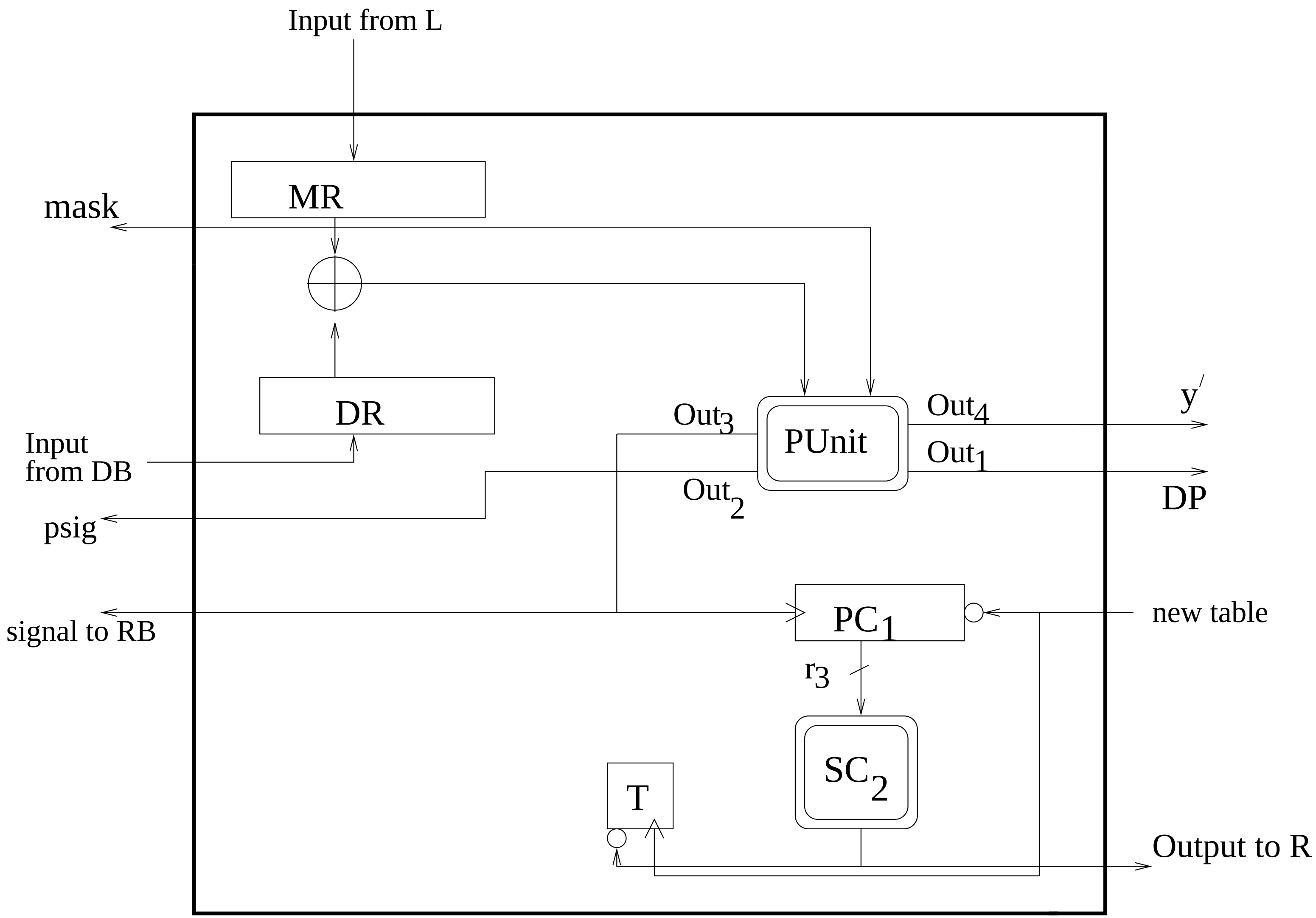
Architecture when f is an iterated round function.

Online Search

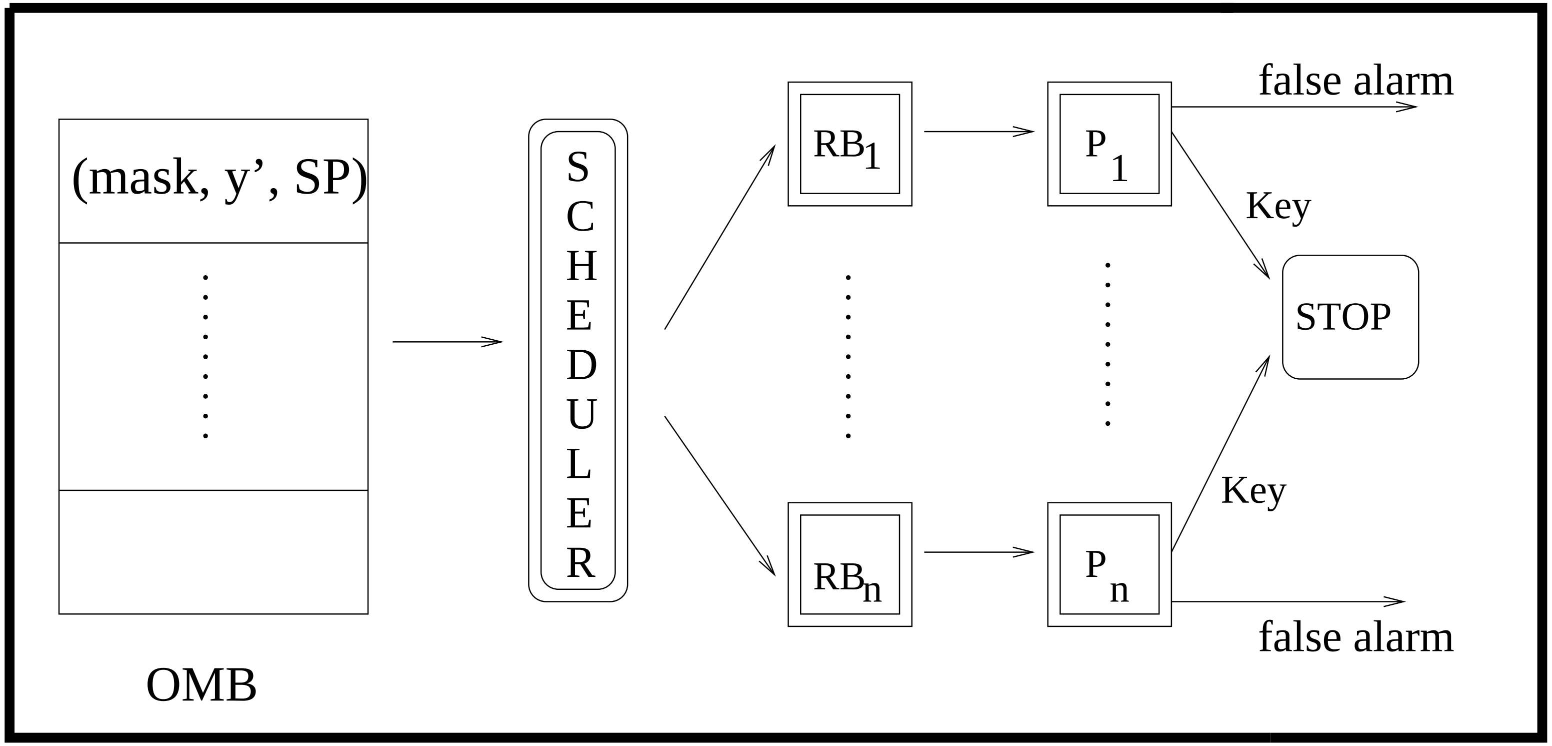


Search in a single table for multiple data points.

Online Search P-processor



Finding the Key



Approximate Cost Analysis

- Assumption: For an s-bit cipher, with $s \leq 128$, the throughput and area will be the same as that of the best AES-128 implementation.
- AES-128 implementation.
  - Good and Benaissa (CHES 2005) proposed a new FPGA design using Xilink Spartan III (XC3S2000).
  - Cost: USD 12.00 per Xilink Spartan III FPGA device.
  - 25 Gbps = 0.2×2^{32} AES-128 encryptions/sec

How Accurate is the Analysis?

- It is an initial study based on one particular architecture.
- Under estimates?
  - Time for DVD load/unload.
  - Time for parallel sorting.
  - Power consumption, circuit failures, ...
- Over estimates?
  - Cost of processor, memory and other components.
  - Speed of encryption.
- Development cost: Very difficult to estimate.

Is the Analysis Meaningful?

- It provides some idea about the relative strength of different key sizes.
- Costs should scale only by small factors.
- Provides an outline of how to perform a cost analysis.
- A more detailed cost analysis will be welcome.
- Different architectures are welcome.

Parameters

- T_{sec}: Pre-computation time in seconds.
- τ_{sec}: Online time in seconds.
- H_p: cost of processors in USD.
- H_m: cost of memory in USD.
- k: number of DVD readers.
- H_w: cost of DVD readers in USD.

Note: Estimates based on Mukhopadhyay-Sarkar (2006).

Table 1: Trade-off for different values of s with $D = 1$.

s	r	m	t	T_{sec}	n	H_p	H_m	k	H_w	τ_{sec}
56	2^{19}	2^{19}	2^{19}	$2^{16.5}$	2^{10}	$2^{13.6}$	2^{19}	$2^{8.5}$	2^{15}	< 1
64	2^{21}	2^{21}	2^{21}	$2^{16.5}$	2^{18}	$2^{21.6}$	2^{21}	2^{12}	$2^{18.5}$	< 1
80	2^{27}	2^{27}	2^{27}	2^{25}	2^{25}	$2^{28.6}$	2^{27}	2^{8}	$2^{14.5}$	< 1
86	2^{29}	2^{29}	2^{29}	2^{25}	2^{31}	$2^{34.6}$	2^{29}	2^{10}	$2^{16.5}$	< 1
96	2^{32}	2^{32}	2^{32}	$2^{38.3}$	2^{28}	2^{32}	2^{32}	1	$2^{6.5}$	80
128	2^{32}	2^{64}	2^{32}	$2^{70.3}$	2^{28}	2^{32}	2^{32}	1	$2^{6.5}$	80

Analysis

- 56-bit and 64-bit keys are completely insecure.
- 80-bit keys: One-year of pre-computation time; cost of USD 500M; online time less than a second.
- 96-bit keys: Pre-computation time is more than 4000 years and cost around USD 1 billion.

Table 2: Trade-off for different values of s and $d = \frac{s}{4}$.

s	r	$m = t$	T_{sec}	n	H_p	H_m	k	H_w	τ_{sec}
80	$2^{6.7}$	$2^{26.7}$	$2^{16.5}$	2^{14}	$2^{17.6}$	$2^{6.7}$	1	$2^{6.5}$	845
86	$2^{6.7}$	$2^{28.6}$	$2^{16.5}$	2^{18}	$2^{21.6}$	$2^{6.7}$	1	$2^{6.5}$	776
96	2^{8}	2^{32}	$2^{16.5}$	2^{26}	$2^{29.6}$	2^{8}	1	$2^{6.5}$	320
96	2^{8}	2^{32}	2^{25}	2^{17}	$2^{20.6}$	2^{8}	1	$2^{6.5}$	$2^{17.3}$
128	2^{11}	2^{43}	2^{25}	2^{41}	$2^{44.6}$	2^{11}	1	$2^{6.5}$	$2^{15.3}$
128	2^{32}	2^{32}	2^{25}	2^{41}	$2^{44.6}$	2^{32}	2^{13}	$2^{19.5}$	$2^{25.3}$
128	2^{32}	2^{32}	2^{38}	2^{28}	2^{32}	2^{32}	1	$2^{6.5}$	$2^{38.3}$

Analysis for $D = 2^{s/4}$

- 80-bit: Attack becomes much easier.
- 96-bit: Attack becomes reasonable.
- 128-bit: At least one of the parameters among (T_{sec}, H_P, H_m) become infeasible.

Currently 128-bit seems to be secure, until a new technological revolution invalidates the analysis performed here.

Desirable Goals

- Exhaustive search.
  - 80-bit keys: Borderline of feasibility.
  - 96-bit keys: Does not look possible.
- TMTO.
  - 56-bit, 64-bit: Methodology validation.
  - 80-bit: With $D = 2^{16}, 2^{20}$. Feasible.
  - 96-bit: With $D = 2^{16}, 2^{20}, 2^{24}$. Ambitious.
- A5/3: 64-bit key, 22-bit IV:
  - Used in practice.
  - TMTO or exhaustive search? Both are doable.

Breaking in the Future

"Today's crypto systems must resist attack by tomorrow's computers. Nanotechnology explores the limits of what we can make."

Ralph Merkle, 15 August 2005
(IACR Distinguished Lecture, CRYPTO).

Processor Technology

- Current technology is CMOS based.
  - Speed of light; clock speed; channel length (of MOSFET).
  - 50 nanometers channel length can produce around 20GHz clock speed.
  - This hits the physical barrier.
- Geobacter.
  - Certain bacteria uses metal for respiration.
  - Can be used to construct conducting nanowires.
  - Quisquater at rump session of Crypto 2005.

Memory Technology

- Blu-ray technology
  - paper disc
  - capability to store several hundred Gb on a single disc.
- Holographic-memory discs:
  - theoretical possibility of one terabyte data storage at an access speed of 120 megabits/sec.
- Non-volatile memory technology.

Interconnection Network

- Connecting many processors to many memory units is a major problem.
- How to bypass the problem?
  - Optical connections?
  - Any other possibilities?

Non-Conventional Technology

- Already proposed attacks on DES.
  - DNA computing – Boneh-Dunworth-Lipton (1995).
  - Photographic techniques – Shamir (1998).
- Other candidates.
  - Quantum computing.
  - Optical computing.

General Idea

- Known plaintext attack. A pair (M, C) is known.
- Define $f(K) = E_K(M)$.
  Goal: Find K.
- Basic idea.
  - For each key K, generate all possible pairs $(K, f(K))$.
  - Pick out the pairs $(K, f(K))$ such that $C = f(K)$.
  - Parallelism: Single instruction multiple data (SIMD) architecture.
  - Exploit denseness of the medium to represent “all possible states”.

General Idea

- Initial state: All possible keys.
- Intermediate state: All possible partial results obtained by encrypting M with all possible keys.
- Final state: All possible ciphertexts obtained by encrypting M with all possible keys.
- Change of state: Effected by performing one step (of DES).
- Total number of operations equals total number of steps of DES.
- Execution of each operation is time consuming.
- Some operations can be executed in parallel, but this kind of parallelism is restricted.

General Idea

- Representation of bit strings in the medium.
- Applying binary operations on bit strings.
  - Applying domain specific operations to implement binary operations.
  - DNA computing: Look-up tables can be implemented.
  - Photographic techniques: Converts look-up tables into Boolean functions.

DNA Computing

- DNA strand: An "oriented" string over the alphabet $\{A, T, C, G\}$.
- Binary string representation.
  - Let length of the string be n.
  - Let $B_i(0), B_i(1)$ be 30-mers used to represent the fact that the ith position is 0 or 1 respectively.
  - Let $S_0, \ldots, S_n$ be 30-mers used as separators.
  - Let $x = x_1 \ldots x_n$. Representation

$$\updownarrow S_0 B_1(x_1) S_1 B_2(x_2) S_2 \ldots S_{n-1} B_n(x_n) S_n$$

  - $S_i, B_i()$'s are distinct and have no long common substrings.

DNA Operations

- Extract: From a solution, separate out strands with a specific substring.
  - Can be used to perform operations on consecutive bits of a binary string.
  - A DES step is either a XOR gate or a table look-up.
  - Suitably defined extract can be used to perform both.
- Amplify: Create duplicates of all DNA strands in the solution.
- Tag: Append a short new sequence onto all strands in the solution.

Analysis

- One litre of solution can contain 10^{17} DNA strands.
- This can just about represent all 2^{56} possible keys of DES.
- Each extract operation is time consuming.
- Estimates.
  - About 10 extract operations in a day.
  - Around 4 months to prepare a solution containing all possible $(K, f(K))$ pairs.
  - Finding K from such a solution can be completed in a few hours.
- Assumes an error-free model.

Photographic Technique

- Each pixel represents a bit.
- Operations:
  - 0-film: Develop a unexposed film.
  - 1-film: Develop a film exposed under strong light.
  - Random film: Photograph a random pattern.
  - NOT: Contact print to another film.
  - NOR: Contact print two stacked films to another film.
  - NAND, XOR: Little more complicated.

Representation of Keys

- Stack 56 random films.
- A vertical line through the films represent a particular key.
- All possible vertical lines represent a set of keys.
- Keys cannot be individually operated upon.
- *Problem:* Number of possible keys can greatly exceed the number of pixels.

Representation of States

- Initial state:
  - Multiple copies of the message.
  - Stack 64 single colour (all 0 or all 1) films.
  - Number of copies equals number of pixels.
- Intermediate state: Result of partial encryption of M under the represented keys.
- Final state: Result of encryption of M under the represented keys.

DES Operations

- Bit permutations and expansions:
  - Easy to perform.
  - Permute films and make copies of films.
- Bitwise XOR.
  - Requires three photographic operations.
- S-box look-up:
  - Can be defined as Boolean functions.
  - Implemented using multi-input NAND, NOR gates and two-input XOR gates.

Can 128-bit Keys be Attacked?

- Storage.
  - This will require the representation of all the 2^{128} keys.
  - DNA Computing: The volume of solution becomes impractical.
  - Photo film: The film area becomes impractical.
- Operations.
  - Each operation (DNA operation or photo printing) is time consuming.
- Conclusion: It is unlikely that AES-128 is going to be meaningfully attacked by such techniques.

A Historical Perspective

This part of the talk is based on “The Code Book” by Simon Singh.

- Enigma: Mechanization of secrecy.
- Extensively used by the Germans during the second world war.
- Polish attack – Rejewski.
  - Exploits double encryption of the message key.
  - A ciphertext only attack.
  - An attack on the mode of operation.
- British attack – Turing.
  - A known plaintext attack.

Enigma Components

- Scramblers.
  - Time varying permutations of the alphabet, i.e., the permutation changes as the encryption proceeds.
  - The order of the scramblers can be changed.
  - Initial setting of the scramblers and their order constitutes the scrambler part of the key.
- Plugboard.
  - A permutation introduced before the scramblers.
  - Not time varying.
  - Settings are part of the key.
- Other components: Reflector, ring.

Mode of Operation

- **Day Key:**
  - Pre-distributed to all concerned parties.
  - Specified the scrambler and plugboard settings.
- **Message Key:**
  - Choose a scrambler setting – three letters.
  - Encrypt the message key twice with the day key.
  - Transmit the double encryption of the message key.
  - Encrypt the message with the message key.

Rejewski's Attack

- **Weakness:** Double encryption of a (unknown) message.
- **Observation:**
  - This defines a permutation of the alphabet which can be constructed only from the ciphertext.
  - Change of plugboard settings gives rise to a conjugate permutation.
  - The cycle structure is *independent* of the plugboard settings.

Attack Techniques

- Divide and conquer.
- Time/Memory trade-off.

Divide and Conquer

- First find the scrambler settings for the day key.
- Next find the plugboard settings for the day key.
  - This is relatively easier.
  - A trial and error method can be used.

Time/Memory Trade-Off

Pre-computation:

- For each possible scrambler setting, compute the corresponding permutation and its cycle structure.
- Store (scrambler setting, cycle structure) in a table.
- Initially required one year hand computation by experts.
- Later mechanized (bombes).

Time/Memory Trade-Off

- **Online search:**
  - Construct the permutation from the ciphertext.
  - Compute its cycle structure.
  - Use table look-up to find the corresponding scrambler setting.
- **Memory:** Required to store the table.

Thank you for your attention!