Cryptographic Engineering

Çetin Kaya Koç

Oregon State University &
Istanbul Commerce University

SBSEG’06 Santos, Brasil

Current Affiliations

- Oregon State University
  On leave, since Sept 2005
- Istanbul Commerce University
  Professor, since Sept 2005
  Information Security Research Center
  Founder & Director
- International research & consulting activities

Research Interests

- □ Research and development in hardware and software realizations of information security and cryptographic systems
- □ Research emphasis on scalable and unified cryptographic processor design, cryptographic design in embedded software, and True Random Number Generators (TRNGs)

Research Applications

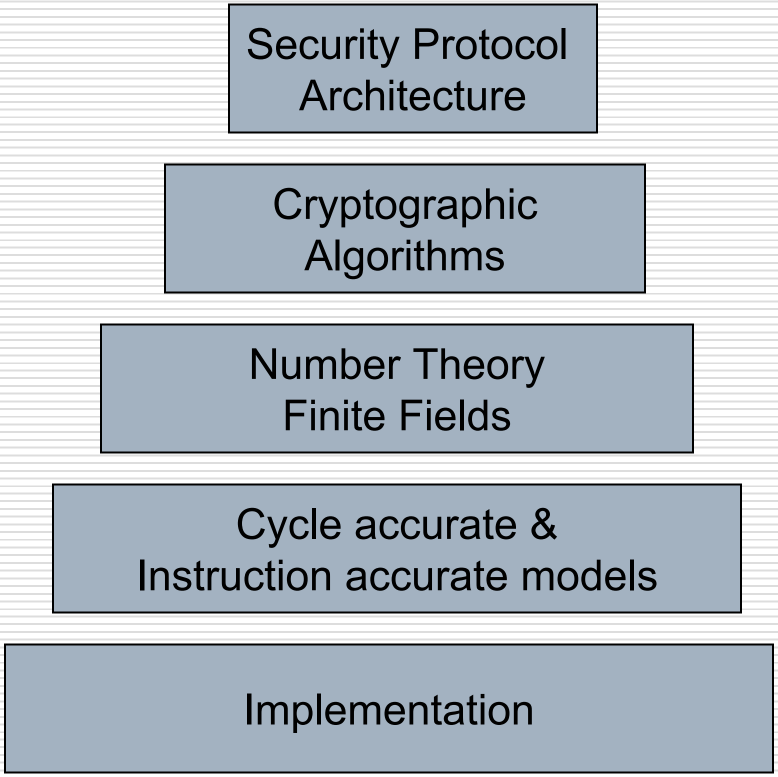
□ High-throughput crypto accelerators for VPNs, SSL servers, and IPSec routers

□ Mobile and resource-constrained devices, smartcards, and cell phones: small PKI, mobile VPN, power-efficient cryptographic modules for encryption and authentication

Cryptographic Engineering

- Cryptographic engineering deals with software and hardware realizations
- Public-key cryptographic algorithms are based on computationally intensive arithmetic and finite-field operations
- Interdisciplinary research area
  - Electrical engineering
  - Computer science
  - Mathematics

Security Pyramid



Security Protocol Architecture	Encryption, integrity and authentication functions, Digital signatures
Cryptographic Algorithms	SHA, RSA, ECC, DES, AES
Number Theory Finite Fields	Galois fields, large primes, special primes
Cycle accurate & Instruction accurate models	Verilog Register Transfer, StrongArm assembly
Implementation	FPGA, Flash, Core, ASIC

Recent Research Activities

- Cryptographic infrastructure work
  - True random number generators
  - Embedded software cryptography
  - Cryptographic coprocessors
- New security products
  - Cryptographic modules
  - Security systems and modules
  - Innovative watermarking

Random Numbers in Cryptography

- Random session key
- RSA prime factors
- Random numbers for DSA
- Zero-knowledge protocols
- Challenge-response protocols
- IV (initializing vectors)

Random Number Generators

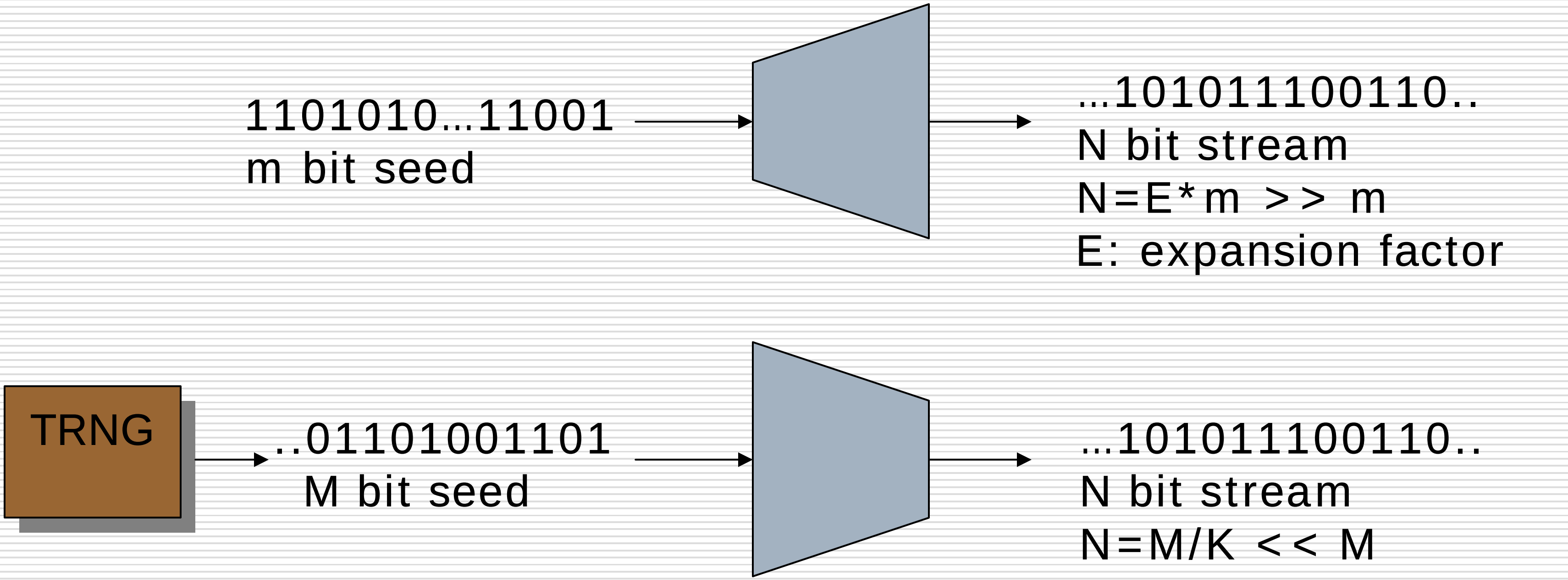
- True (physical) random number generators (TRNGs)
- Deterministic random number generators (DRNGs) – output is completely determined by the seed
- Hybrid generators – refresh their seed regularly, e.g., by exploiting user’s interaction, mouse movements, key strokes, or register values

Requirements

Requirements depend essentially on the application

- R1: The random numbers should have good statistical properties
- R2: The knowledge of subsequences of random numbers should not enable to compute predecessors or successors or to guess them with non-negligible probability

Pseudo versus True Random Number Generators



1101010…11001
m bit seed

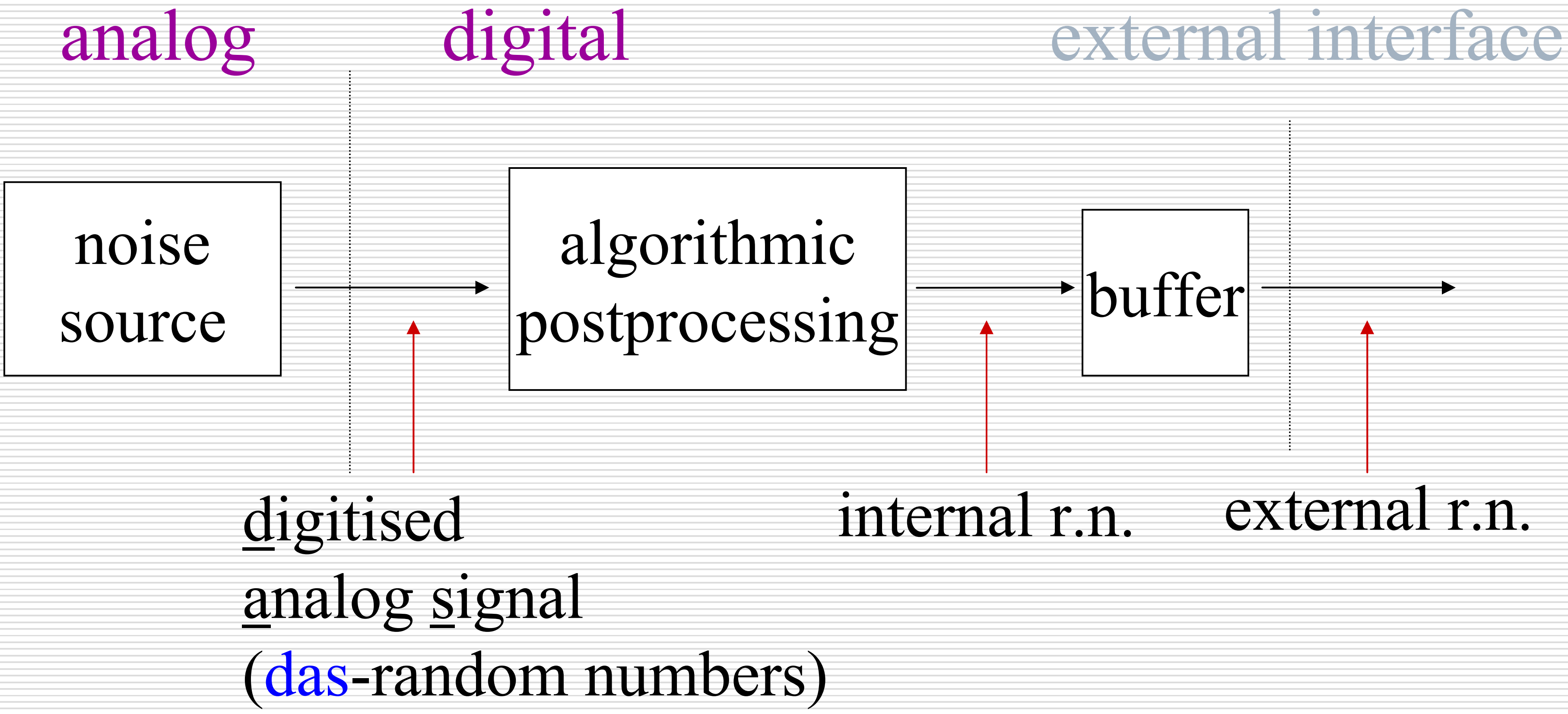
…101011100110..
N bit stream
N=E*m >> m
E: expansion factor

TRNG

..011010011101
M bit seed

…101011100110..
N bit stream
N=M/K << M
K: compression factor

TRNG General Design



analog

digital

external interface

noise source

algorithmic postprocessing

buffer

digitised
analog signal
(das-random numbers)

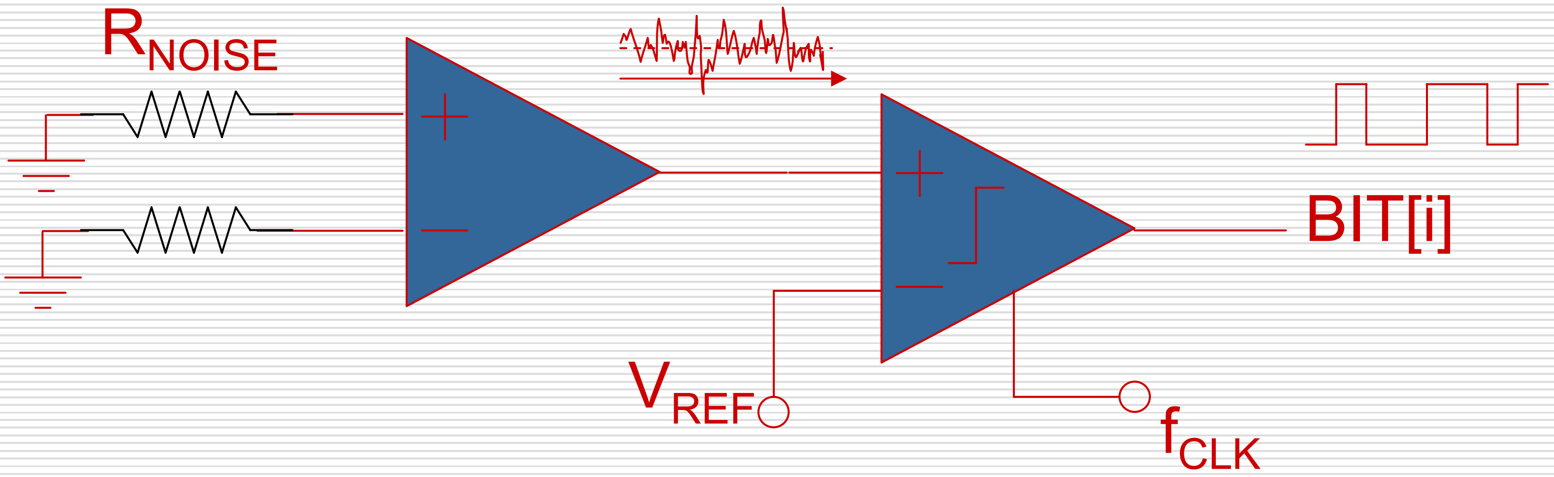
internal r.n.

external r.n.

TRNG References

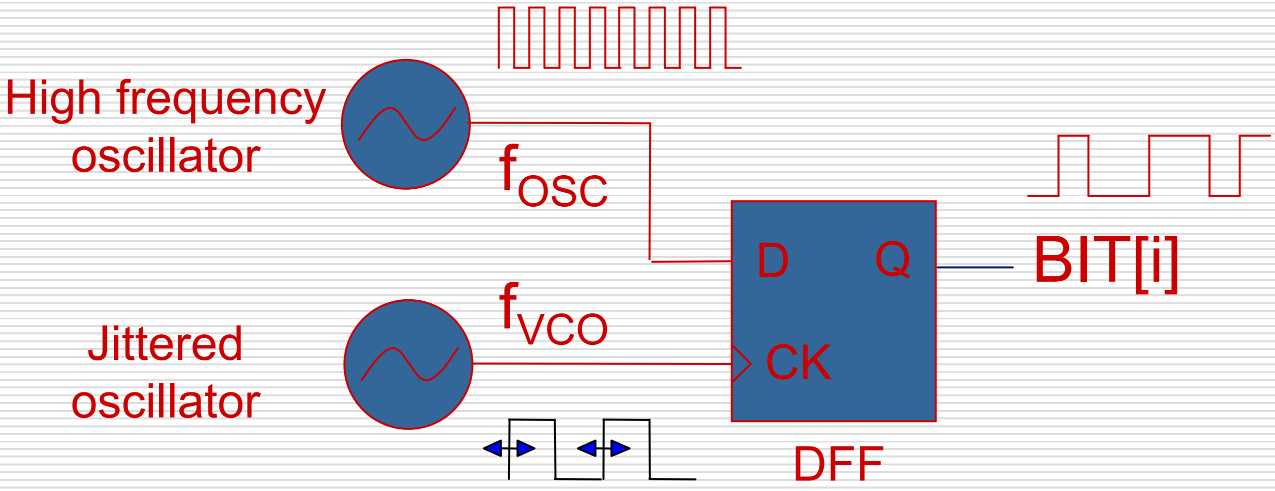
- V. Bagini, M. Bucci, “A design of a reliable true random number generator for cryptographic applications”, Proc. CHES 99, Lecture Notes in Computer Sciences 1717, Springer-Verlag, Heidelberg, Germany, pp. 204-218, 1999.
- M. Bucci, L. Germani, R. Luzzi, A. Trifiletti, M. Varanonuovo, “A high speed oscillator-based truly IC random number source for cryptographic applications on smart card”, IEEE Trans. Computers, Special Issue on Cryptographic Hardware and Embedded Systems, pp.403-409, April 2003.
- W.T. Wolman, J.A. Connelly, A.B. Dowlatabadi, “An integrated analog/digital random noise source”, IEEE Trans. Circuits and Systems I, vol. 44, no. 6, pp. 521-528, June 1997.
- B. Jun, P. Kocher, “The Intel random number generator”, Cryptography Research Inc., white paper prepared for Intel Corp., April 1999, at http://www.cryptography.com/resources/whitepapers/IntelRNG.pdf
- T. Stojanovski, L. Kocarev, “Chaos-based random number generators – Part I: Analysis”, IEEE Trans. Circuits and Systems I, vol. 48, no. 3, pp. 281-288, March 2001.
- E. Trichina, M. Bucci, D. De Seta, R. Luzzi, “Supplemental cryptographic hardware for smart cards”, IEEE Micro, vol. 21, no. 6, 2001.

Direct Amplification



W.T. Wolman, J.A. Connelly, A.B. Dowlatabadi, "An integrated analog/digital random noise source", **IEEE Trans. Circuits and Systems I**, vol. 44, no. 6, pp. 521-528, June 1997.

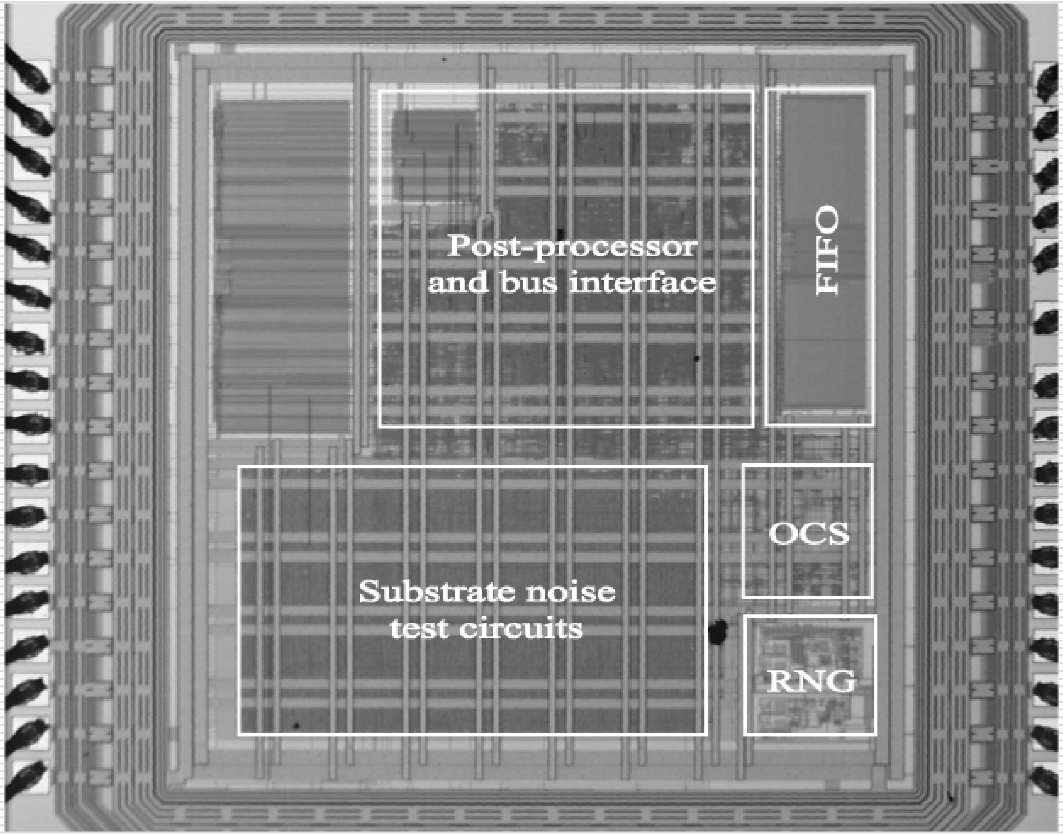
Oscillator Sampling



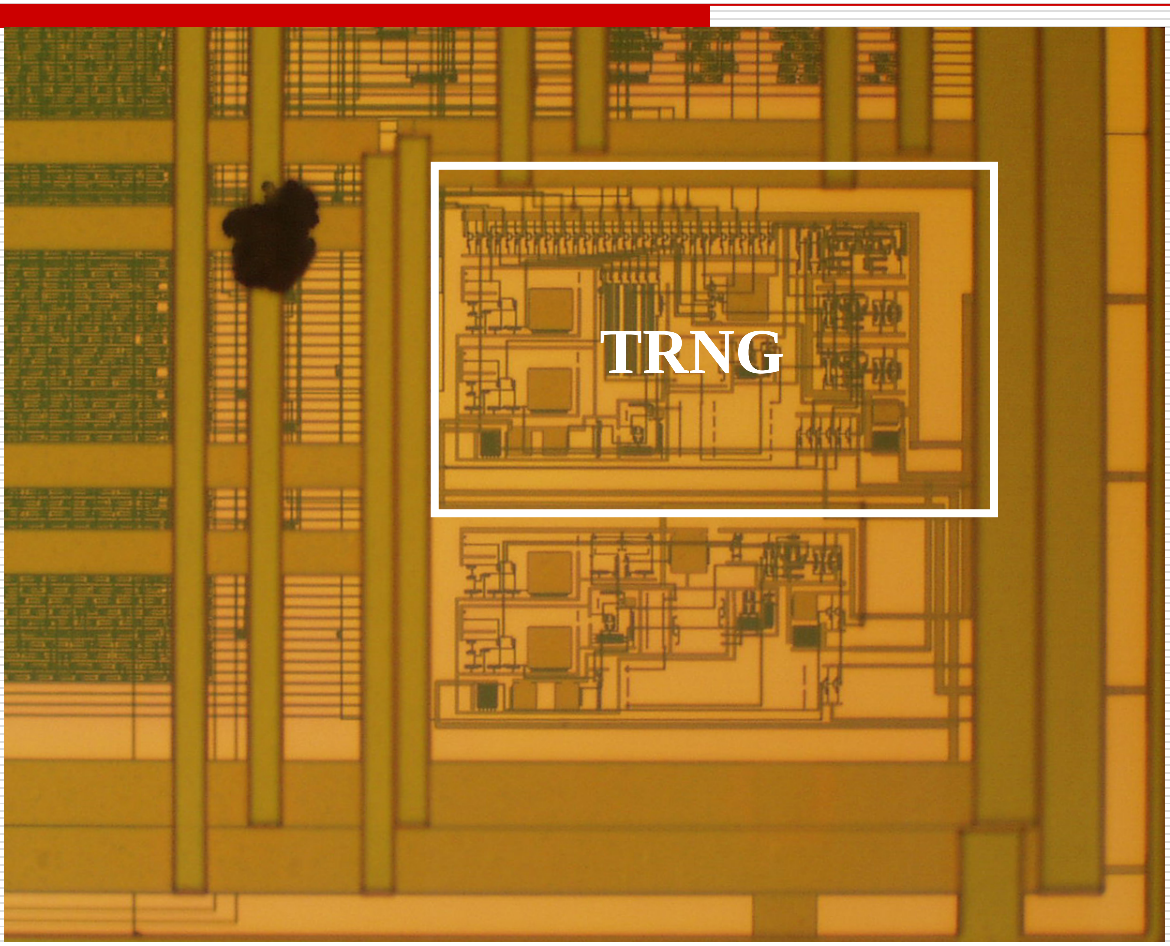
B. Jun, P. Kocher, "The Intel random number generator",
Cryptography Research Inc., White paper prepared for Intel Corp., April 1999, at
http://www.cryptography.com/resources/whitepapers/IntelRNG.pdf

TRNG Test Chips

- ❑ Process:
  TSMC 0.18μm
- ❑ Chip area:
  0.025mm^2
  (220μm×116μm)
- ❑ Power supply:
  3.3V/1.8V
- ❑ Power consumption:
  ≈ 3.6mW


Post-processor and bus interface
FIFO
OCS
Substrate noise test circuits
RNG

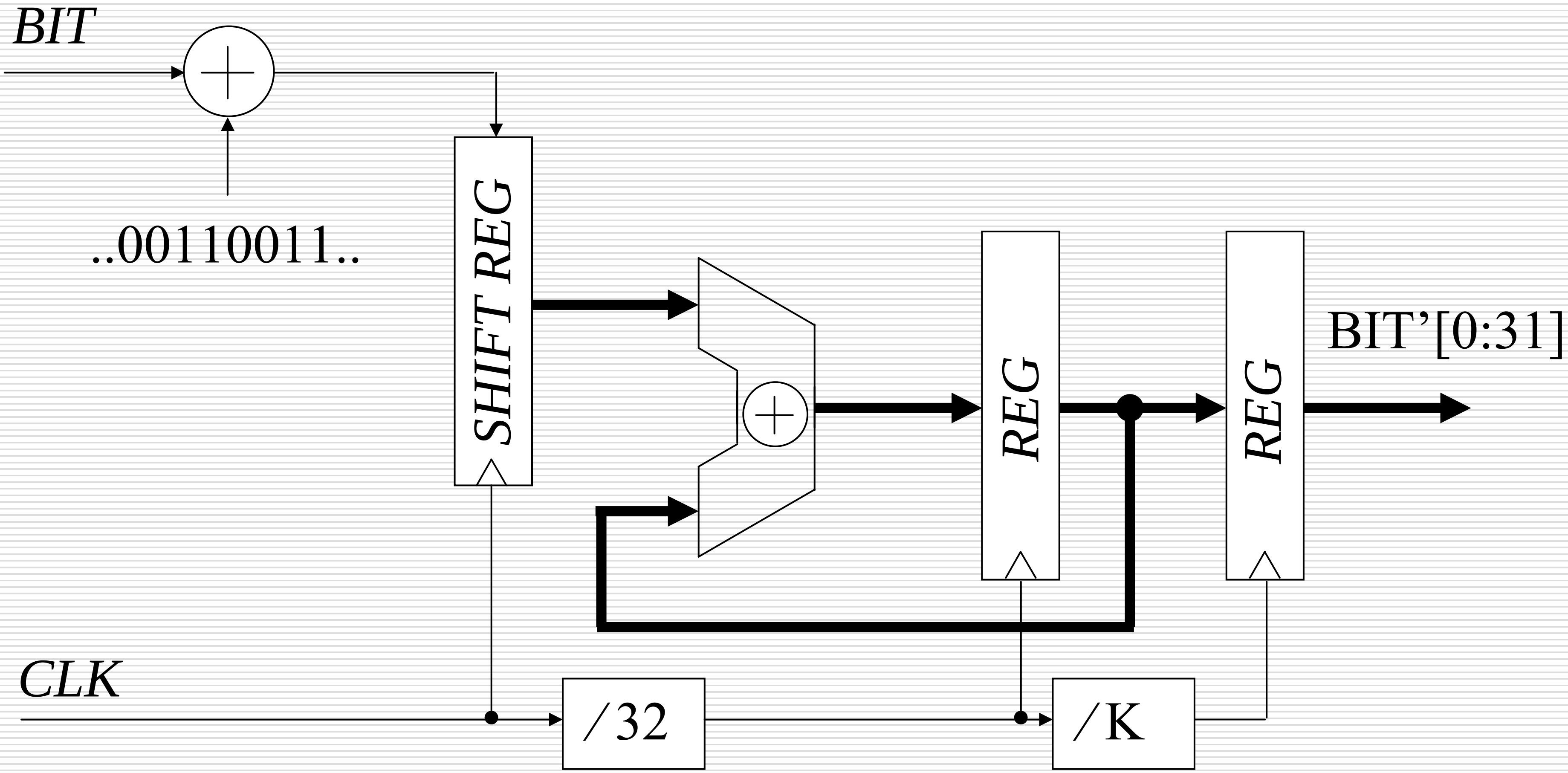
TRNG Test Chip Detail



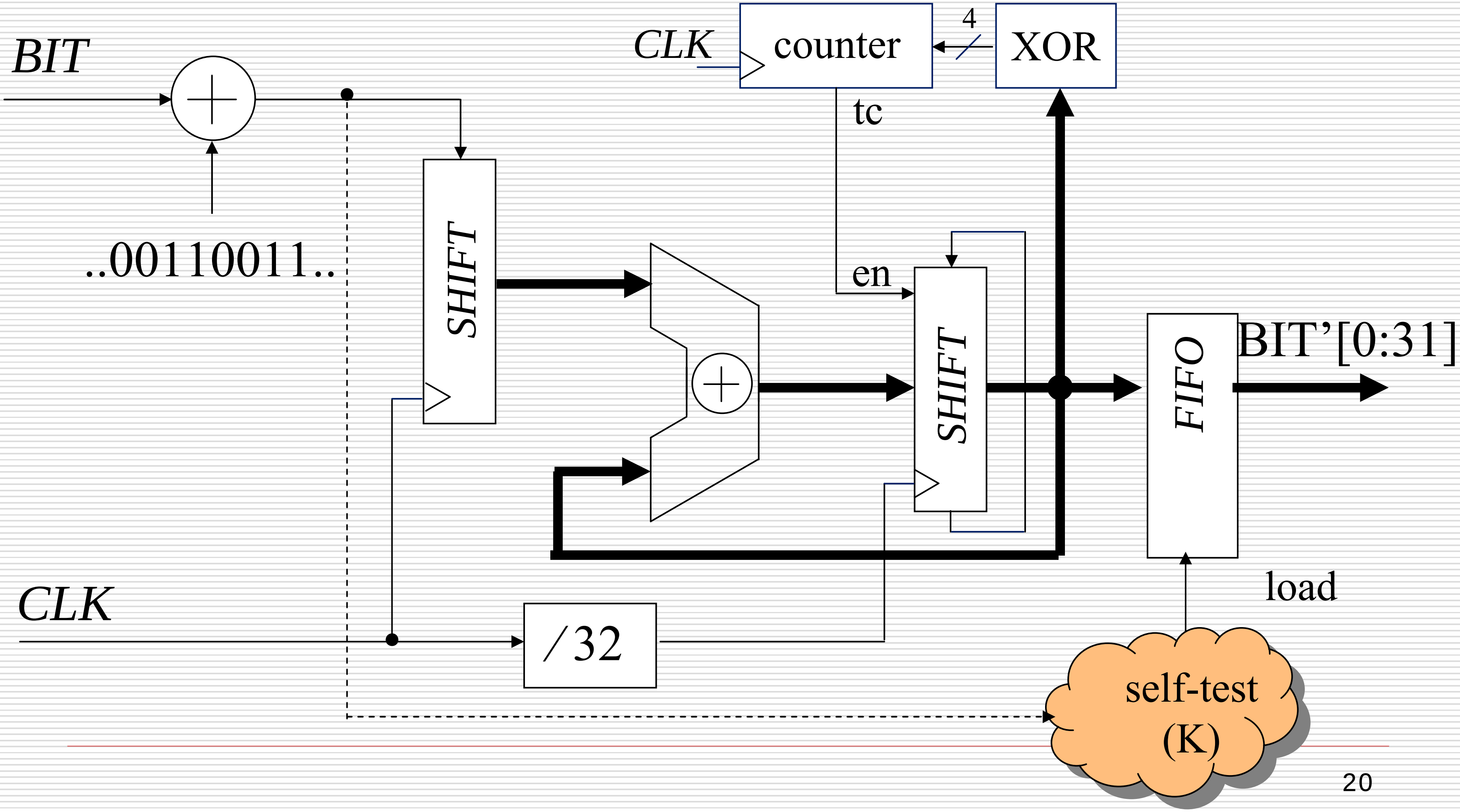
Randomness Tests

- Maurer's Test
  U. M. Maurer, "A Universal Statistical Test for Random Bit Generators", *Journal of Cryptology*, vol. 5, no. 2, 1992, pp. 89-105.
- Diehard
  G. Marsaglia, "A current view of random number generator", *Proc. Computer Science Statistics: 16th Symp. Interface*, Keynote Address, 1984.
- NIST Tests
  NIST Special Publication 800-22, "A statistical test suite for random and pseudorandom number generator for cryptographic application", September 2000.
- FIPS Tests
  "FIPS 140-1, Security requirements for cryptographic modules", Federal Information Processing Standards Publication 140-1. U.S. Department of Commerce/NIST, National Technical Information Service, Springfield, VA, 1994.

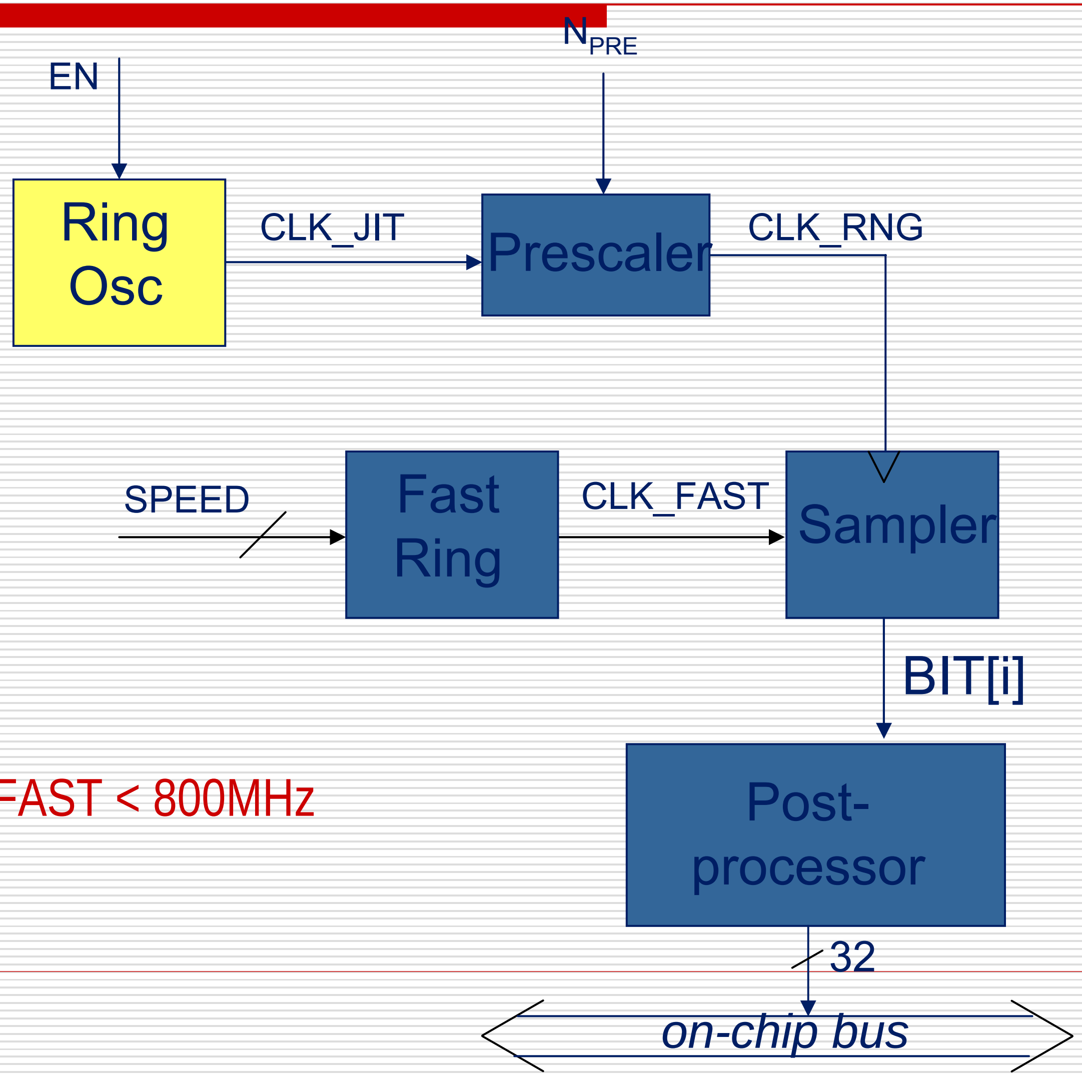
TRNG – Postprocessor1



TRNG – Postprocessor2

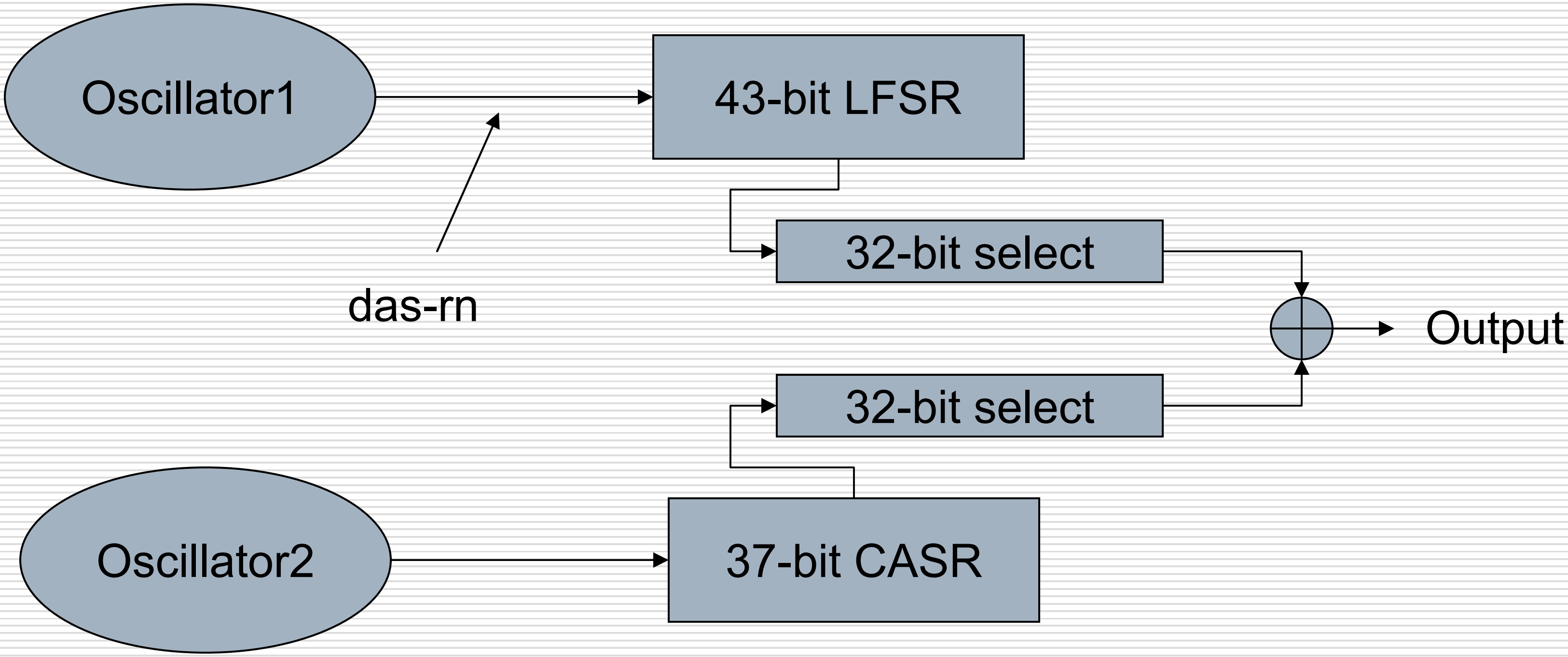


A TRNG Architecture



CLK_FAST < 800MHz

Another Example of TRNG



TRNGs in Operation: Problems

- Total breakdown of the noise source
- Aging effects
- Tolerances of components

Tests

Tests	Aim
Tot-test	Shall detect a total breakdown of the noise source very quickly
Startup test	Shall ensure the functionality of the TRNG at the start
Online test	Shall detect non-tolerable weakness or deterioration of the quality of random numbers

Evaluation of TRNGs

- ITSEC (Information Security Evaluation Criteria) and CC (Common Criteria) **do not specify** any uniform evaluation criteria for random number generators
- NIST **does not offer** any standard method for evaluating TRNGs (no FIPS for such purpose)
- The only TRNG evaluation standard in the world: AIS 31 (German standard)

AIS 31

- Published by BSI (Bundesamt fuer Sicherheit in der Informationstechnik) on Sep 2001

  http://www.bsi.bund.de/zertifiz/zert/interpr/ais_cc.htm

- Provides clear evaluation criteria for TRNGs
- Distinguishes between two functionality classes
  - P1 - less sensitive (challenge-response)
  - P2 – sensitive (key generation)

3 Prototypes

- RNG1: High speed amplification-based
- RNG2: High quality oscillator-based
- RNG3: Full digital (standard cells)
- RNG4: ....

- Working on several TRNGs at the same time and select the best one: in terms of cost, chip area requirements, quality of randomness, robustness, and reliability

TRNG Provable Quality

- The overall design to be approved by international bodies
- Extensive analytical and statistical tests to be performed internally
- Tests under various attack scenarios
- Create robust, trusted TRNGs for across the board systems

TRNG Project Plan

- Design of noise source generators
- Implementation
- Design of post-processors
- Randomness testing
- Validation
- Decision

Cryptographic Coprocessor

- Design of several cryptographic hardware modules
- A unified design for a coprocessor family to be used in several different products
- Provides scalability for future upgrades
- Area-time tradeoffs for environments with different constraints and requirements

ECC/RSA Hardware Design

- Two types of finite fields are more commonly used in many real-world applications
  - Prime fields: GF(p)
  - Binary extension fields: GF(2^k)
- These fields have dissimilar properties
- Different design possibilities
- Different implementations on specialized hardware

Unified (Dual-Field) Arithmetic

- A unified hardware design methodology is possible for both fields since
  - The elements of either field are represented using almost the same data structures
  - The algorithms for basic arithmetic operations in both fields have structural similarities, i.e., the steps of the algorithms are nearly identical

Benefits of Unified Arithmetic

- Low manufacturing cost
- Compatibility

However, the design needs to be

- Scalable
- Fast and parallel
- Impartial (does not favor one prime against another or one irreducible polynomial against another)

Montgomery Arithmetic

Montgomery multiplication is the right choice since

- □ Suitable for unified design and works for both
- □ Scalable
- □ Parallelizable
- □ Suitable for pipelining
- □ Impartial

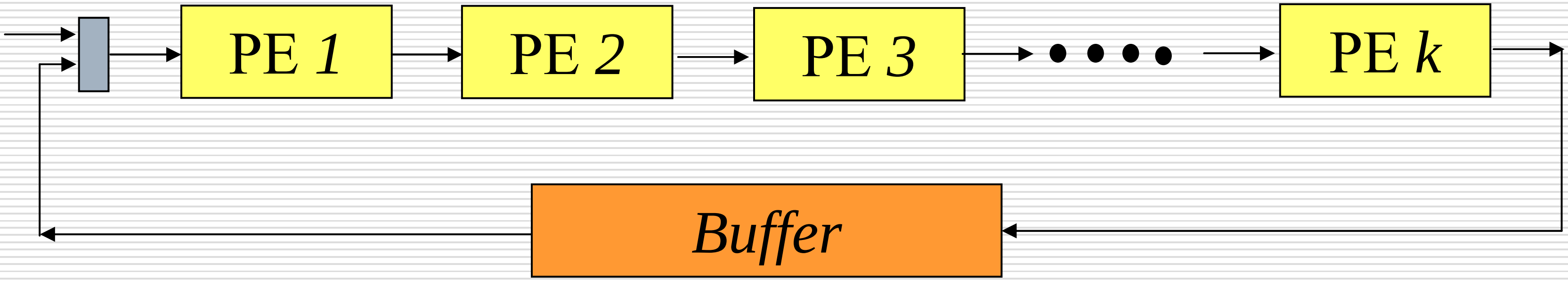
Scalability

- An architecture is scalable if

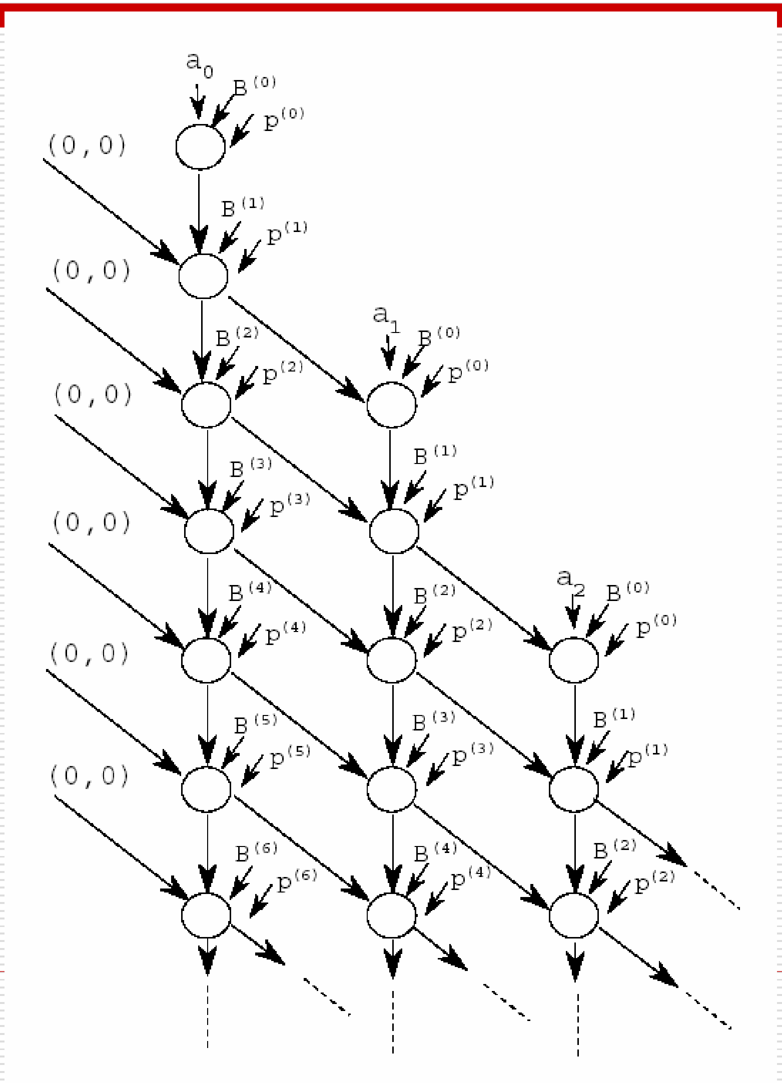
  it can be reused or replicated in order to generate long-precision results independently of the data path precision for which it was originally designed

- Application-specific architectures are generally limited by the data path for which they were designed

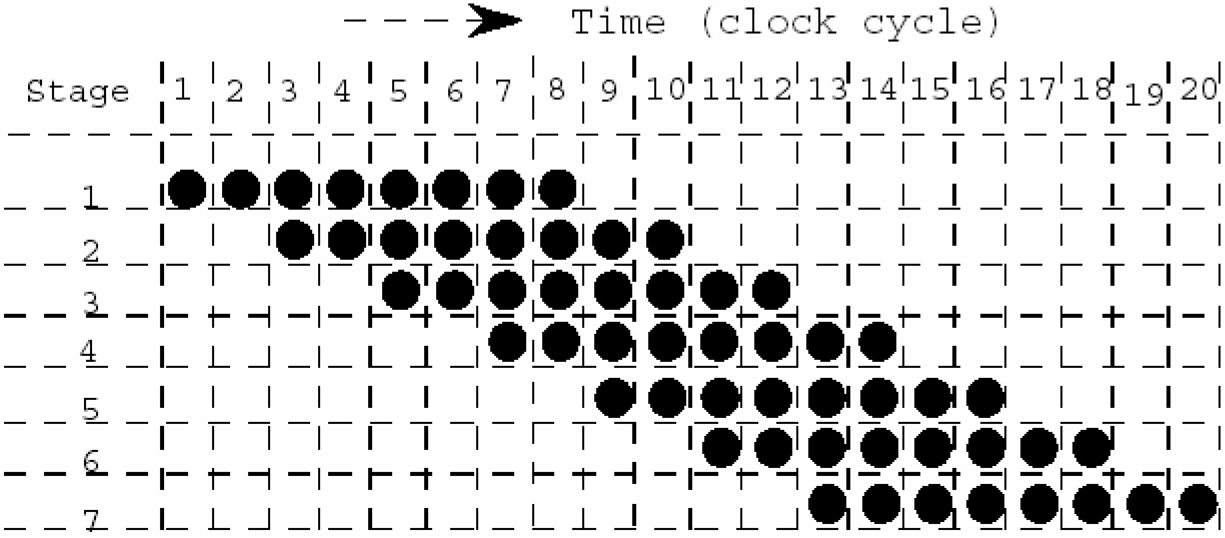
Scalable Architecture



Dependency Graph of Montgomery



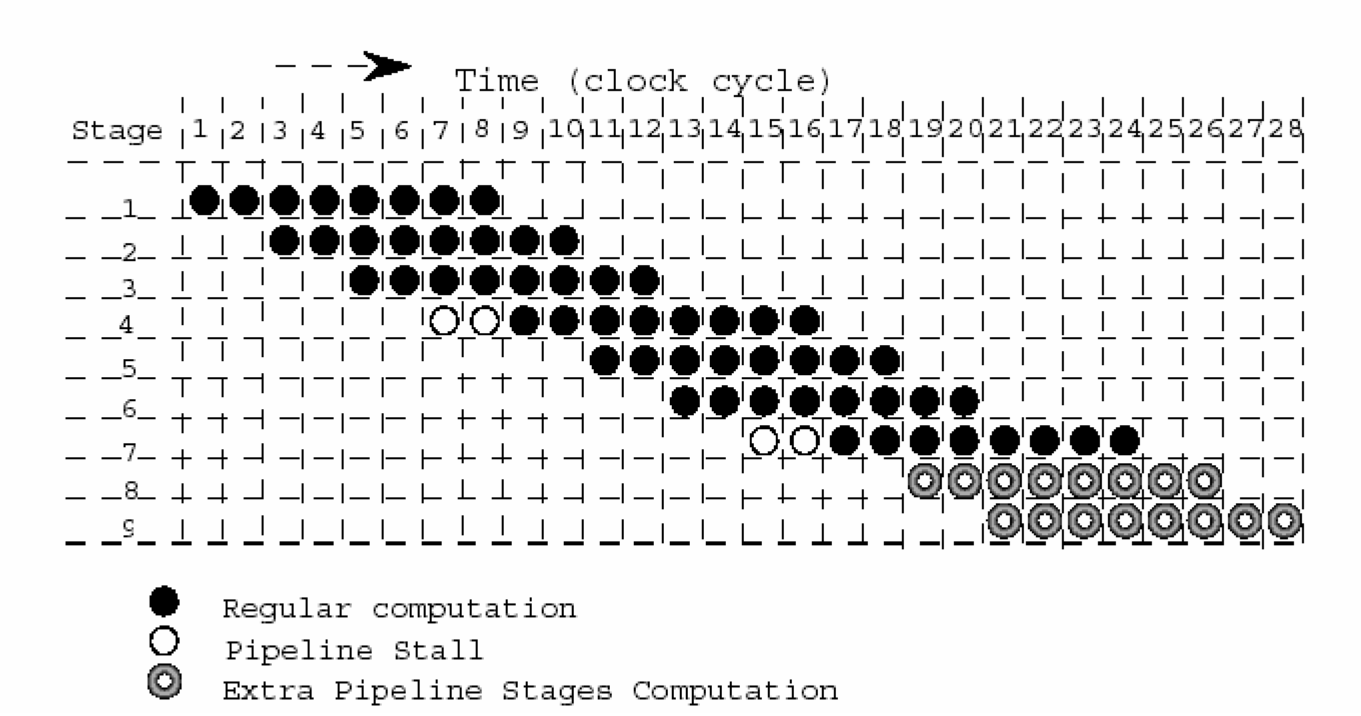
Pipelined Computation



Stage	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	●	●	●	●	●	●	●	●												
2			●	●	●	●	●	●	●	●										
3					●	●	●	●	●	●	●	●								
4							●	●	●	●	●	●	●	●						
5									●	●	●	●	●	●	●	●				
6											●	●	●	●	●	●	●	●		
7													●	●	●	●	●	●	●	●

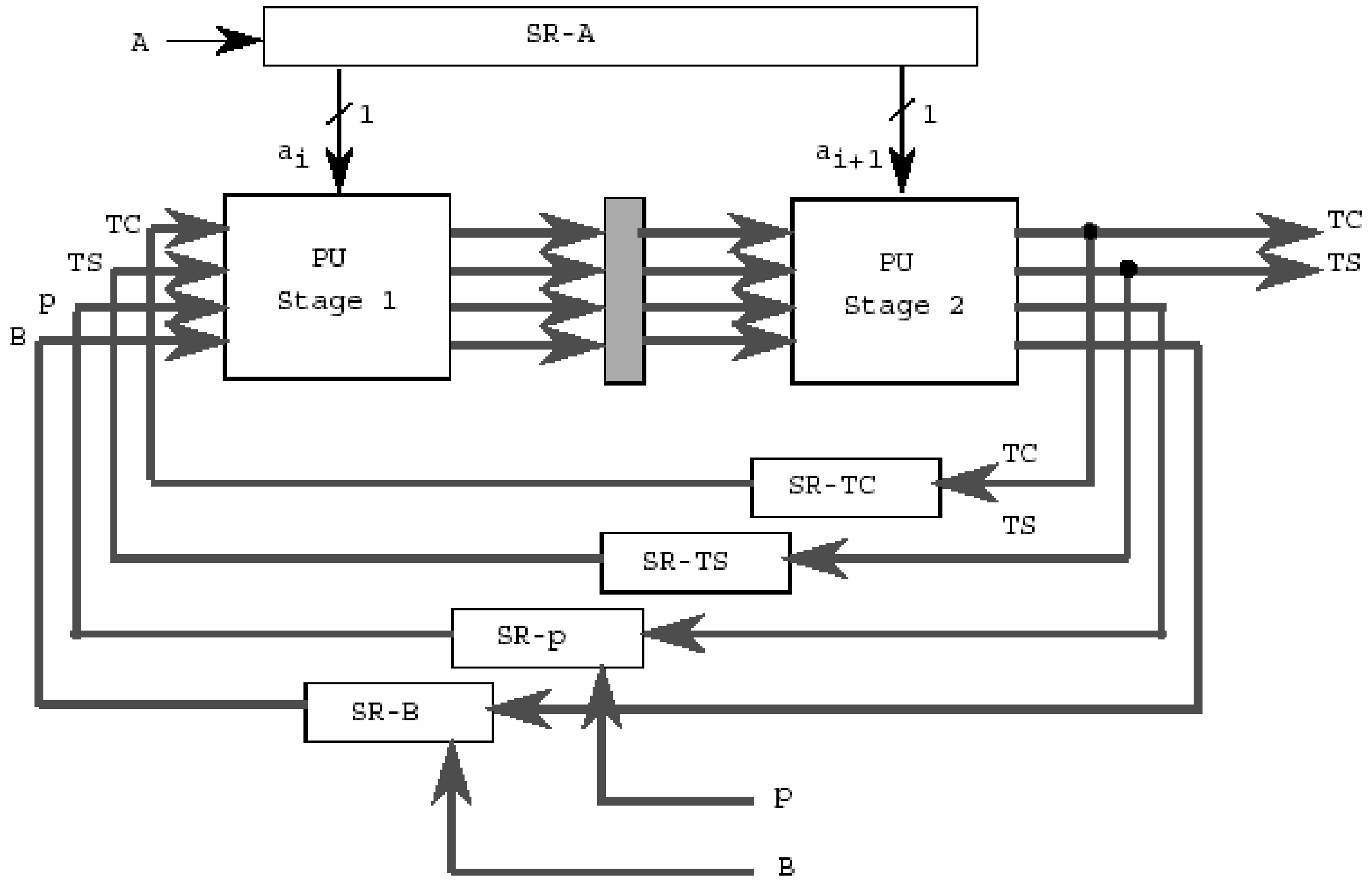
An example of pipeline computation
for 7 bit operands where word-length is 1 bit

Pipeline Stalls

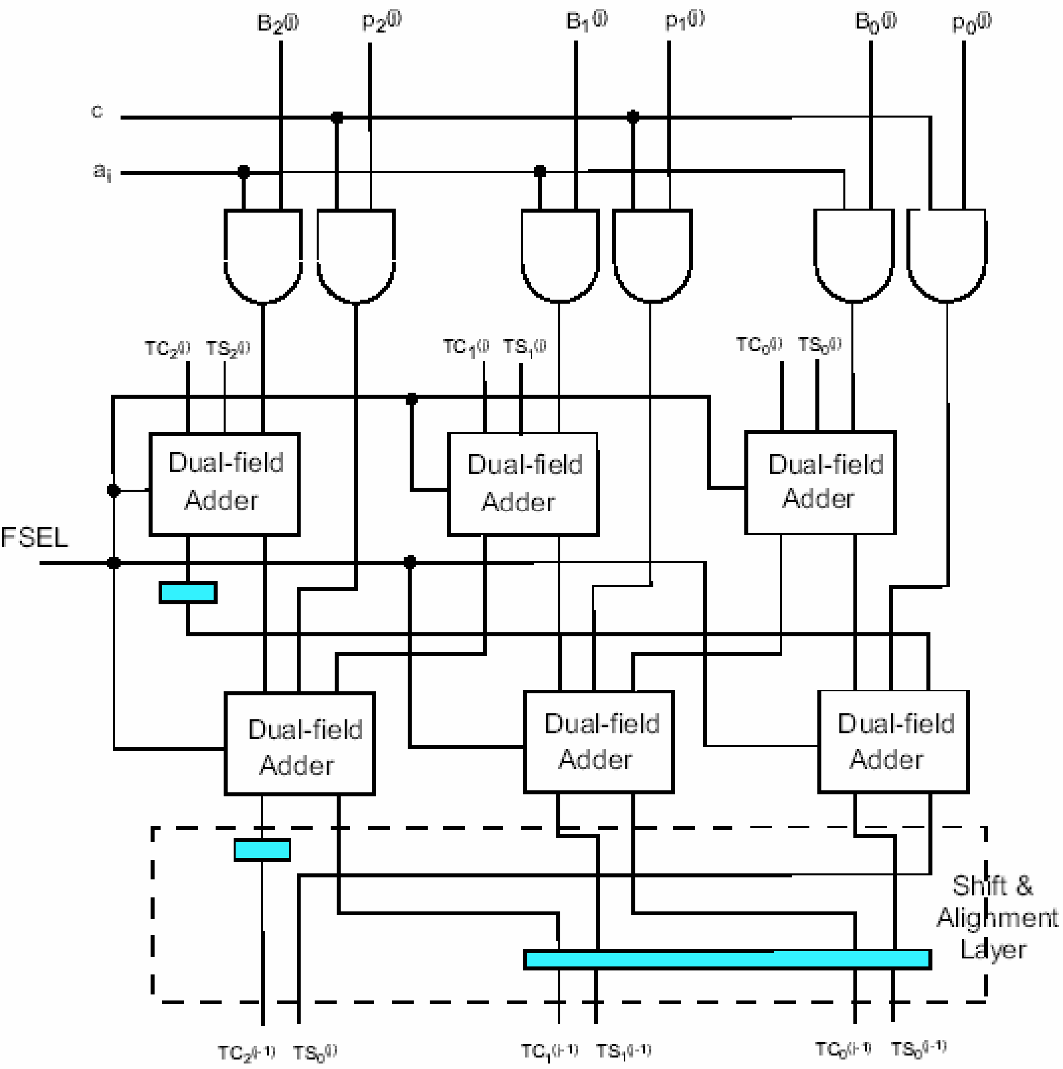


Pipeline stalls when fewer processing units are available, here m=7, w=1, k=3

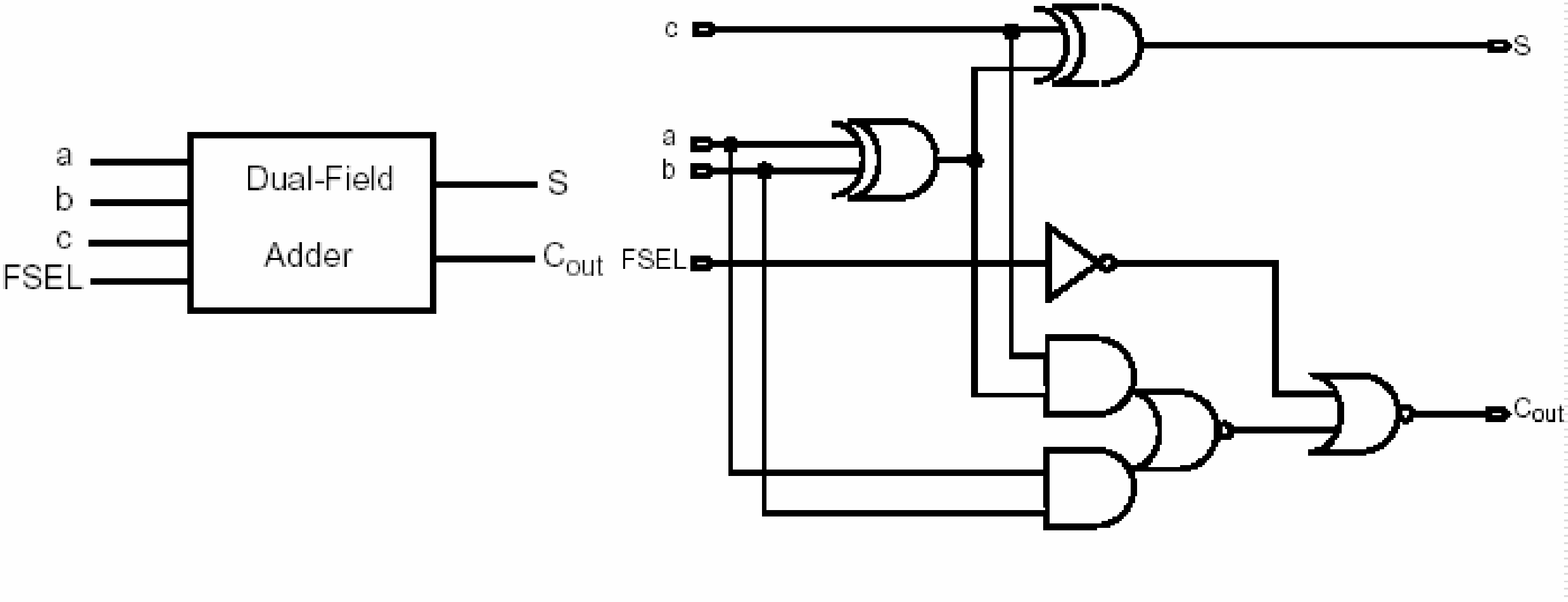
Pipeline Organization with 2 Units



3-bit Processing Unit



Dual-field Adder



(a) Dual-Field Adder

(b) Synthesized circuit by Mentor

Synthesis Results

- PU is synthesized with Mentor Tools using 1.2 micron CMOS technology
- 2-input NAND gate takes 0.94 chip area
- Unified field area (w) = 48.5w
- Only GF(p) area (w) = 47.2w
- Latch area = 8.32w
- Total Area for k-stage pipeline = 56.82kw – 8.32w
- Propagation time is 11ns
- Clock frequency 90MHz

Security Products Objectives

- Creation of a Road Map
- Design several security architectures (architectural scenarios) with different kinds of security objectives/levels
- Create documents for detailed security requirements for different terminals (cell phone, PDA, smartcard, etc.)
- Create security solutions with generic properties satisfying a wide-range of requirements

Security Classification

No security	Minimal Security (no crypto)	Basic Security (simple security features)	Advanced Security (security features, crypto functions, certificates)	High-End Security (advanced security features, full crypto, certificates)
First US analog cell phones	Simple PC security Software against viruses, etc	Loyalty, metering, basic GSM, identification	Banking (debit, credit), access, m/e-commerce, healthcare	Banking, e-purse, PKI, pay-TV, multifunction cards, DRM, trustworthy computing

Security Needs

- Tamperproofness (which level)
- Security placement (which level)
- Cryptographic performance
- What type of application
- Overall system security
- IMEI, SIM lock, etc. protection
- Immunization and counter-measures against side-channel attacks

Attacks and Countermeasures

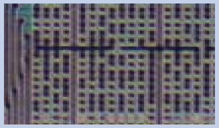
Hardware Attacks

Optical Analysis

Memory Preparation

Test-Mode Attack

Hardware Re-Engineering

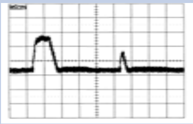


Electrical Attacks

Probing

Electrical Visualization

Physical Stressing

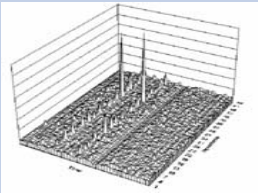


Analytical Attacks

Fault Attack

Timing Attack

Power Attack



Countermeasures

Security memory cells (ROM, EEPROM)

Hidden layout techniques

Memory scrambling

Address scrambling

Proprietary CPU kernel

Various sensors and filters

Hardware encryption

Randomizing features

Test-mode locking mechanism

And some more techniques

Trusted Phone Applications

- The use of cell phone as a storage and/or applicator of smartcards
  - Concept: VSC (virtual smartcard)
  - A software approach to create multifunction smartcards
  - A methodology for SSO (single-sign-on)
  - Needs to confirm with ISO standards
  - Interface through a USB or similar port

Trusted Phone Applications

- The use of cell phone as a rolling key generator (similar to RSA SecureID)
  - Currently exists as a separate device
  - Integrated with the phone using a complete silicon solution
  - Visual interface with the user
  - Needs to work with enterprise desktop authentication software

What Future Will Bring

- Which crypto algorithms are needed in future
  - Design of flexible, patent-free crypto modules
  - Allow doubling of key and block sizes (scalability)
- New side-channel attacks
  - Can we win against the new attacks
  - Use of proven security mechanisms
- Can we make provably secure chips?
  - Formal methods for attack characterization
  - Secure functionality by formal verification of 16-bit and 32-bit CPUs
  - Side-channel attack resistance and tamper resistance

What Future Will Bring

- Innovative watermarking techniques
  - Use of physical one-way functions and physical signatures
  - Support from the device technologies
  - Creation of other physical one-way functions supported by RFIDs
- Innovative use of error-detecting and error-correcting codes
- Ubiquitous and pervasive computing
  - Solutions should be offered in silicon
  - Low-cost, cheap solutions for basic security functionalities
  - Leadership and innovation are the most important traits at this juncture