

SEA for internet-0: a Scalable Encryption Algorithm for Small Embedded Applications

F.-X. Standaert, G. Piret,
N. Gershenfeld, J.-J. Quisquater



why we need crypto for internet-0

- **identification** : use encryption in a specific way
- **authentication** : use encryption in a specific way
- **encryption** : indeed

- thus asking encryption algorithm with *security*, *scalability* (uses, processors, ...), *small footprint*, maybe trading it.



The competition

- Triple-DES:
- AES:
- TEA (Tiny Encryption Algorithm), FSE 1994
- Yuval: “Reinventing the Travois”, FSE 1997
 - no scalability
 - do we only need one standard encryption algorithm?
(NB: need of something like tinySSL for UDP)



Specifications

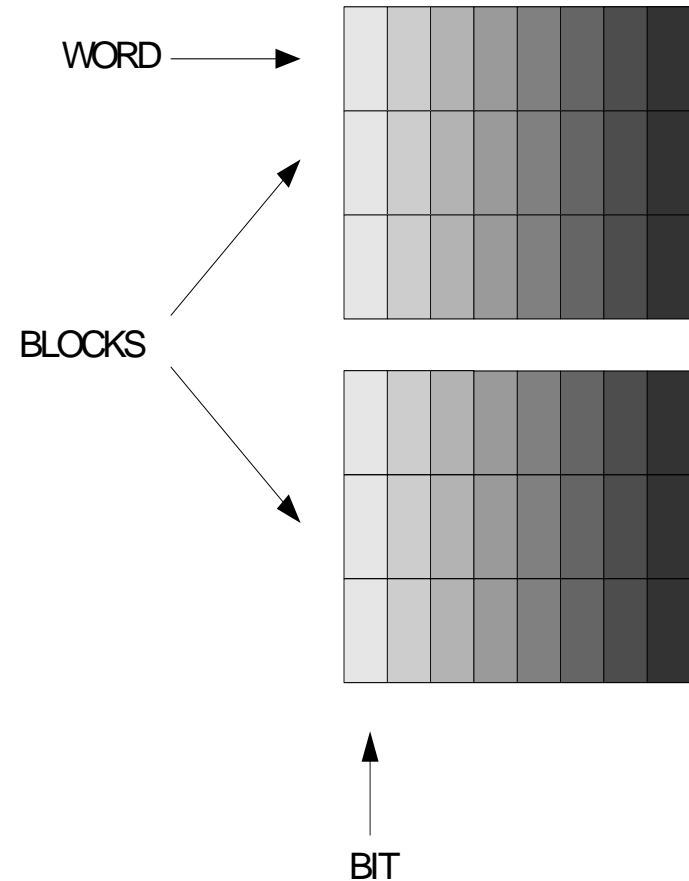
- n : plaintext size, key size.
- b : processor (or word) size.
- $n_b = \frac{n}{2b}$: number of words per Feistel branch.
- n_r : number of block cipher rounds.

→ $SEA_{n,b}$

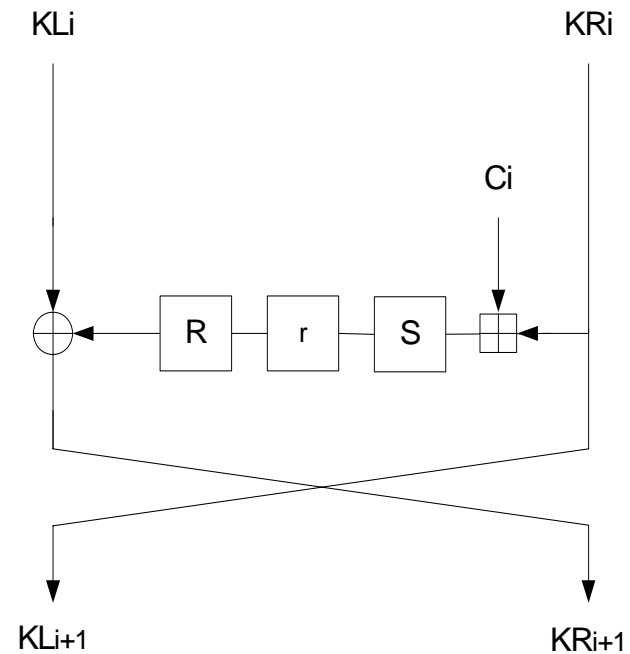
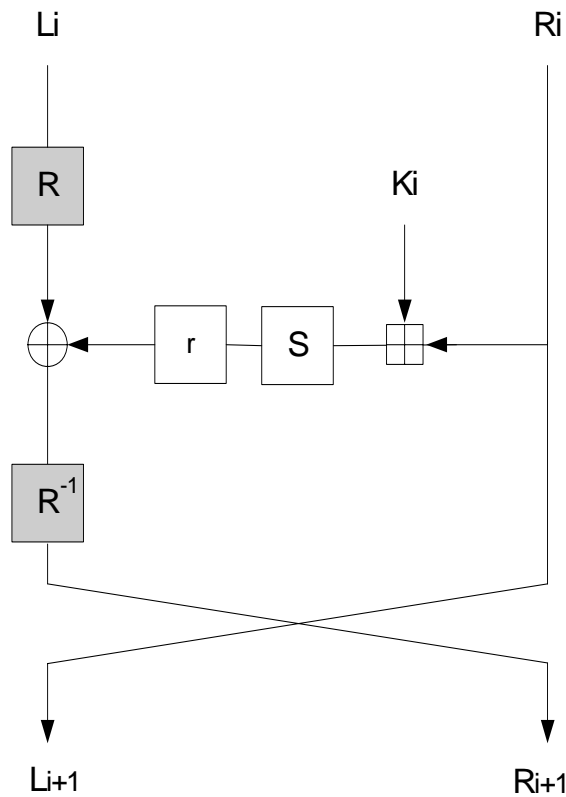


Basic operations

1. Bitwise XOR \oplus
2. Substitution box S
3. Word rotation R
4. Bit rotation r
5. Addition mod 2^b \boxplus



Round and key round



Performances

- Pseudo-assembly code provided
- Operation counts easy
- Only a few instructions required:
 - Arithmetic and logic
 - Branch instructions
 - Comparisons, load from RAM, store in RAM



It yields...

	# ram	# regs.	code size (ops.)	implementation time (ops.)
$SEA_{n,b}$	$4n_b$	$n_b + 3$	$31n_b + 36$	$(n_r - 1) \times (22n_b + 29) + 20n_b + 18$

Table 1. Performance evaluation of $SEA_{n,b}$ (encryption + decryption).

Algorithm	E/D	Device	# ram	# regs.	code size	# clock cycles	# cycles \times code size
$SEA_{96,8}$	yes	Atmel ATtiny	1	32	386	17 745	$6849 \cdot 10^3$
$SEA_{192,32}$	yes	ARM (risc-32)	6	12	420	27 059	$11\,364 \cdot 10^3$
Rijndael [19]	no	ARM (risc-32)	16	12	1404	2889	$4056 \cdot 10^3$
$SEA_{128,32}$	yes	ARM (risc-32)	6	12	280	18 039	$5050 \cdot 10^3$

Table 2. Comparisons: the code size is expressed in bytes. The results of $SEA_{128,32}$ were obtained by multiplying the code size and number of cycles of $SEA_{192,32}$ by $2/3$, since 128 is not a multiple of 6.



ASIC and FPGA

- ASIC: 6800 gates 271 Mbits/s (250 MHz)
- FPGA: 400 slices 240 Mbit/s (240 MHz)



Conclusions

- Efficient combination of encryption/decryption
- Low code size
- Low memory requirements (RAM + regs)
- Typical performances: a few milliseconds and a few hundreds bytes of ROM
- More efficient for large bus sizes
- Compared to the AES: trades time for space
 - May be reasonable in recent controllers

