

# KATAN & KTANTAN — A Family of Small and Efficient Hardware-Oriented Block Ciphers

Orr Dunkelman

Département d'Informatique  
École Normale Supérieure

France Telecom Chaire

Joint work with Christophe De Cannière and  
Miroslav Knežević



# Outline

- 1 Introduction
  - Why the AES is not Suitable for Low-end Devices
  - Other Solutions for Constrained Environments
- 2 Design Goals
  - What do you expect of a cipher?
  - Really Low-end Devices
- 3 Building Blocks
  - Bivium
  - LFSR Counter
  - Two Round Functions
- 4 The KATAN Block Ciphers
  - Key Schedule Algorithm
  - Taps
- 5 The KTANTAN Block Ciphers
- 6 Performance Analysis
- 7 Security Analysis

# Why the AES is not Suitable for Low-end Devices

- ▶ The AES was selected at the end of a very long development effort.
- ▶ It is deemed as the block cipher to answer all symmetric keys needs in the 21st century.

# Why the AES is not Suitable for Low-end Devices

- ▶ The AES was selected at the end of a very long development effort.
- ▶ It is deemed as the block cipher to answer all symmetric keys needs in the 21st century.

**Is it?**

# Why the AES is not Suitable for Low-end Devices

- ▶ The AES was selected at the end of a very long development effort.
- ▶ It is deemed as the block cipher to answer all symmetric keys needs in the 21st century.

## Is it?

- ▶ AES can be efficient in hardware, but the smallest implementation is 3.1 K gates.
- ▶ AES may not be suitable in constrained environments due to other considerations.
- ▶ Cache-timing attacks may render AES unsuitable to some software environments.

# Other Solutions for Constrained Environments

- ▶ Stream ciphers

# Other Solutions for Constrained Environments

- ▶ Stream ciphers
  - ▶ To ensure security, the internal state must be twice the size of the key.
  - ▶ No good methodology on how to design these.

# Other Solutions for Constrained Environments

- ▶ Stream ciphers
  - ▶ To ensure security, the internal state must be twice the size of the key.
  - ▶ No good methodology on how to design these.
- ▶ Block ciphers

# Other Solutions for Constrained Environments

- ▶ Stream ciphers
  - ▶ To ensure security, the internal state must be twice the size of the key.
  - ▶ No good methodology on how to design these.
- ▶ Block ciphers
  - ▶ HIGHT, mCrypton, DESL, DES, PRESENT.

# Other Solutions for Constrained Environments

- ▶ Stream ciphers
  - ▶ To ensure security, the internal state must be twice the size of the key.
  - ▶ No good methodology on how to design these.
- ▶ Block ciphers
  - ▶ HIGHT, mCrypton, DESL, DES, PRESENT.
  - ▶ Can we do better?

# Design Goals

- ▶ Secure block cipher
  - ▶ Differential/Linear cryptanalysis — very large safety margins.
  - ▶ Related-Key/Slide attacks — foil using no constants.
  - ▶ Related-Key differentials — do not exist.
- ▶ Efficient block cipher
  - ▶ Small foot-print
  - ▶ Low power consumption
  - ▶ Reasonable performance (+ possible speed-ups)

# Really Low-end Devices

Does an RFID tag really need to support key agility?

- ▶ Some low-end devices have one key throughout their life cycle.
- ▶ Why to waste good gates on their key-agility features?

# Really Low-end Devices

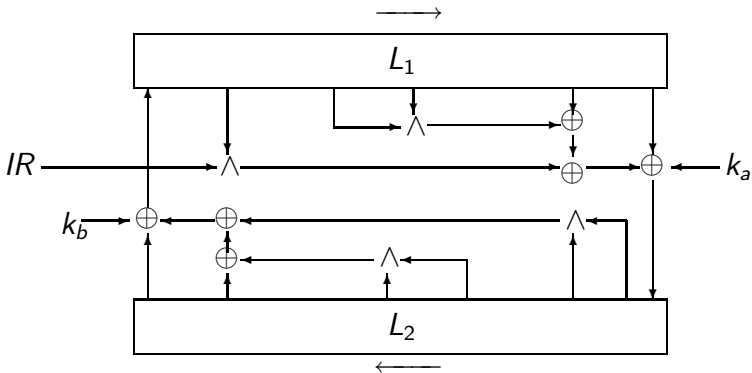
Does an RFID tag really need to support key agility?

- ▶ Some low-end devices have one key throughout their life cycle.
- ▶ Why to waste good gates on their key-agility features?
- ▶ Some low-end devices are going to encrypt very little data throughout their life cycles.
- ▶ Why to waste good gates on their ability to encrypt more messages than that?

# The Basic Building Blocks

- ▶ Bivium (Trivium with two registers) in a block cipher mode.
- ▶ LFSR counts rounds (rather than a counter).
- ▶ Two round functions (the one to use is controlled by a bit of the LFSR).

# The Basic Building Blocks — Bivium



# The Basic Building Blocks — LFSR counter

- ▶ When counting the number of rounds, you can use a counter.

# The Basic Building Blocks — LFSR counter

- ▶ When counting the number of rounds, you can use a counter.
- ▶  $n$ -bit counter  $\Rightarrow n - 1$ -long carry chain.
- ▶  $n$ -bit LFSR — a bit of control.

# The Basic Building Blocks — LFSR counter

- ▶ When counting the number of rounds, you can use a counter.
- ▶  $n$ -bit counter  $\Rightarrow n - 1$ -long carry chain.
- ▶  $n$ -bit LFSR — a bit of control.
- ▶ Checking end conditions: overflow in counter (carry chain longer) or special internal state (LFSR/counter).

# The Basic Building Blocks — Two Round Functions

- ▶  $IR$  is a bit which defines which of the two round functions to use.
- ▶ It toggles between two functions.

# The Basic Building Blocks — Two Round Functions

- ▶  $IR$  is a bit which defines which of the two round functions to use.
- ▶ It toggles between two functions.
- ▶ Prevents any slide attacks, and increases diffusion.
- ▶ Uses the MSB of from the LFSR to pick the function (another advantage of an LFSR over counter).

# The KATAN Block Ciphers

- ▶ KATAN has 3 flavors: KATAN-32, KATAN-48, KATAN-64.
- ▶ Block size: 32/48/64 bits.
- ▶ Key size: 80 bits.
- ▶ Share the same key schedule algorithm, and the only difference in the encryption — tap positions.
- ▶ Share same number of rounds — 254 (LFSR of 8 positions).

# The KATAN Block Ciphers — Key Schedule

- ▶ Key is loaded into an 80-bit LFSR.
- ▶ Each round, the LFSR is clocked twice, and two bits are selected  $k_a$  and  $K_b$ .
- ▶ (Polynomial:  $x^{80} + x^{61} + x^{50} + x^{13} + 1$ ).

# The KATAN Block Ciphers — Tap Positions

Cipher	$ L_1 $	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$
KATAN32	13	12	7	8	5	3
KATAN48	19	18	12	15	7	6
KATAN64	25	24	15	20	11	9

Cipher	$y_1$	$y_2$	$y_3$	$y_4$	$y_5$	$y_6$
KATAN32	18	7	12	10	8	3
KATAN48	28	19	21	13	15	6
KATAN64	38	25	33	21	14	9

# The KATAN Block Ciphers — Final Touches

- ▶ KATAN32 is clocked once in each round.

# The KATAN Block Ciphers — Final Touches

- ▶ KATAN32 is clocked once in each round.
- ▶ KATAN48 is clocked twice in each round ( $k_a$  and  $k_b$  are the same for both invocations).

# The KATAN Block Ciphers — Final Touches

- ▶ KATAN32 is clocked once in each round.
- ▶ KATAN48 is clocked twice in each round ( $k_a$  and  $k_b$  are the same for both invocations).
- ▶ KATAN64 is clocked three times in each round ( $k_a$  and  $k_b$  are the same for the tree invocations).

# The KTANTAN Block Ciphers

- ▶ KTANTAN has 3 flavors: KTANTAN-32, KTANTAN-48, KTANTAN-64.
- ▶ Block size: 32/48/64 bits.
- ▶ Key size: 80 bits.
- ▶ KATAN- $n$  and KTANTAN- $n$  are the same up to key schedule.
- ▶ In KTANTAN, the key is burnt into the device and cannot be changed.

# The KTANTAN Block Ciphers — Burnt Key?!?

- ▶ Many devices are deployed in such a manner that the key is initialized once and never changed.
- ▶ Maintaining key agility is not important.

# The KTANTAN Block Ciphers — Burnt Key?!?

- ▶ Many devices are deployed in such a manner that the key is initialized once and never changed.
- ▶ Maintaining key agility is not important.
- ▶ And it saves about 80 bits of memory + 4 XOR gates for the feed back.

# The KTANTAN Block Ciphers — Burnt Key?!?

- ▶ Many devices are deployed in such a manner that the key is initialized once and never changed.
- ▶ Maintaining key agility is not important.
- ▶ And it saves about 80 bits of memory + 4 XOR gates for the feed back.
- ▶ For such devices, we allow the key to be burnt once, and the key schedule algorithm is composed of picking the next bit.

# The KTANTAN Block Ciphers — Key Schedule

- ▶ Main problem — related-key and slide attacks.
- ▶ Solution A — two round functions, prevents slide attacks.
- ▶ Solution B — divide the key into 5 words of 16 bits, pick bits in a nonlinear manner.

# The KTANTAN Block Ciphers — Key Schedule

- ▶ Main problem — related-key and slide attacks.
- ▶ Solution A — two round functions, prevents slide attacks.
- ▶ Solution B — divide the key into 5 words of 16 bits, pick bits in a nonlinear manner.
- ▶ Specifically, let  $K = w_4 || w_3 || w_2 || w_1 || w_0$ ,  $T = T_7 \dots T_0$  be the round-counter LFSR, set:

$$a_i = \text{MUX16to1}(w_i, T_7 T_6 T_5 T_4)$$

$$k_a = \overline{T_3} \cdot \overline{T_2} \cdot (a_0) \oplus (T_3 \vee T_2) \cdot \text{MUX4to1}(a_4 a_3 a_2 a_1, T_1 T_0),$$

$$k_b = \overline{T_3} \cdot T_2 \cdot (a_4) \oplus (T_3 \vee \overline{T_2}) \cdot \text{MUX4to1}(a_3 a_2 a_1 a_0, \overline{T_1 T_0})$$

# Performance Analysis — A Story of a Memory Bit

- ▶ A standard D flip-flop uses 8 NAND gates.
- ▶ It can be squeezed down a bit in the real layout.

# Performance Analysis — A Story of a Memory Bit

- ▶ A standard D flip-flop uses 8 NAND gates.
- ▶ It can be squeezed down a bit in the real layout.
- ▶ Many just copy the standard flip-flop of their library.

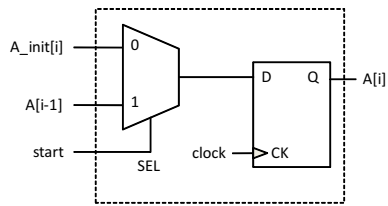
# Performance Analysis — A Story of a Memory Bit

- ▶ A standard D flip-flop uses 8 NAND gates.
- ▶ It can be squeezed down a bit in the real layout.
- ▶ Many just copy the standard flip-flop of their library.
- ▶ Not so good idea, especially as the internal state of low-end devices takes most of the area!

# Performance Analysis — A Story of a Memory Bit

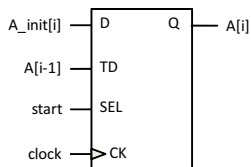
- ▶ A standard D flip-flop uses 8 NAND gates.
- ▶ It can be squeezed down a bit in the real layout.
- ▶ Many just copy the standard flip-flop of their library.
- ▶ Not so good idea, especially as the internal state of low-end devices takes most of the area! We use a scan flip-flop (6.25 GE/bit).

## Performance Analysis — A Story of a Memory Bit



7.25 ~ 13.75 GE

≡



6.25 ~ 11.75 GE

# Implementation Results

- ▶ We used *fsc01\_d\_sc\_tc* 0.13 $\mu$ m family standard cell library tailored for UMC's 0.13 $\mu$ m Low Leakage process.
- ▶ Aimed for lowest possible foot print (but also explored more throughput in exchange for more hardware).

# Performance Analysis — Implementation Results (cont.)

Cipher	Block	Key	Area	GE/bit	Throughput	Logic
AES-128	128	128	3100	5.8	0.08	0.13 $\mu\text{m}$
DES	64	56	2309 <sup>†</sup>	12.19	44.4	0.18 $\mu\text{m}$
DESL	64	56	1848 <sup>†</sup>	12.19	44.4	0.18 $\mu\text{m}$
PRESENT-80	64	80	1570	6	200	0.18 $\mu\text{m}$
PRESENT-80	64	80	1000	N/A	11.4	0.35 $\mu\text{m}$
Grain	1	80	1294	7.25	100	0.13 $\mu\text{m}$
Trivium	1	80	749	2 <sup>◇</sup>	100 <sup>‡</sup>	0.35 $\mu\text{m}$
KATAN32	32	80	802	6.25	12.5	0.13 $\mu\text{m}$
KATAN48	48	80	902	6.25	18.8	0.13 $\mu\text{m}$
KATAN64	64	80	1008	6.25	25.1	0.13 $\mu\text{m}$
KTANTAN32	32	80	462	6.25	12.5	0.13 $\mu\text{m}$
KTANTAN48	48	80	562	6.25	18.8	0.13 $\mu\text{m}$
KTANTAN64	64	80	662	6.25	25.1	0.13 $\mu\text{m}$

# Performance Analysis — Implementation Results (cont.)

Cipher	Block	Key	Area	GE/bit	Throughput	Logic
KATAN32	32	80	874	6.25	25	0.13 $\mu\text{m}$
KATAN32	32	80	908	6.25	37.5	0.13 $\mu\text{m}$
KATAN48	48	80	947	6.25	37.6	0.13 $\mu\text{m}$
KATAN48	48	80	1008	6.25	56.4	0.13 $\mu\text{m}$
KATAN64	64	80	1048	6.25	50.2	0.13 $\mu\text{m}$
KATAN64	64	80	1104	6.25	75.3	0.13 $\mu\text{m}$
KTANTAN32	32	80	673	6.25	25	0.13 $\mu\text{m}$
KTANTAN32	32	80	893	6.25	37.5	0.13 $\mu\text{m}$
KTANTAN48	48	80	772	6.25	37.6	0.13 $\mu\text{m}$
KTANTAN48	48	80	988	6.25	56.4	0.13 $\mu\text{m}$
KTANTAN64	64	80	874	6.25	50.2	0.13 $\mu\text{m}$
KTANTAN64	64	80	1088	6.25	75.3	0.13 $\mu\text{m}$

# Security Analysis — Security Targets

- ▶ Differential cryptanalysis — no differential characteristics with probability  $2^{-n}$  for 127 rounds.
- ▶ Linear cryptanalysis — no approximation with bias  $2^{-n/2}$  for 127 rounds.
- ▶ No related-key/slide attacks.
- ▶ No related-key differentials (probability at most  $2^{-n}$  for the entire cipher).
- ▶ No algebraic-based attacks.

# Security Analysis — Differential Cryptanalysis

- ▶ Computer-aided search for the various round combinations and all block sizes.
- ▶ KATAN32: Best 42-round char. has prob. at most  $2^{-11}$ .
- ▶ KATAN48: Best 43-round char. has prob. at most  $2^{-18}$ .
- ▶ KATAN64: Best 37-round char. has prob. at most  $2^{-20}$ .

# Security Analysis — Differential Cryptanalysis

- ▶ Computer-aided search for the various round combinations and all block sizes.
- ▶ KATAN32: Best 42-round char. has prob. at most  $2^{-11}$ .
- ▶ KATAN48: Best 43-round char. has prob. at most  $2^{-18}$ .
- ▶ KATAN64: Best 37-round char. has prob. at most  $2^{-20}$ .
- ▶ This also proves that all the differential-based attacks fail (boomerang, rectangle).

# Security Analysis — Linear Cryptanalysis

- ▶ Computer-aided search for the various round combinations and all block sizes.
- ▶ KATAN32: Best 42-round approx. has prob. at most  $2^{-6}$ .
- ▶ KATAN48: Best 43-round char. has prob. at most  $2^{-10}$ .
- ▶ KATAN64: Best 37-round char. has prob. at most  $2^{-11}$ .
- ▶ This also proves that differential-linear attacks fail.

# Security Analysis — Slide/Related-Key Attacks

- ▶ Usually these are prevented using constants.
- ▶ In the case of KATAN/KTANTAN — solved by the irregular function use.
- ▶ In KATAN — the key “changes” (no slide).
- ▶ In KTANTAN — order of subkey bits not linear.

# Security Analysis — Related-Key Differentials

- ▶ No good methodology for that.

# Security Analysis — Related-Key Differentials

- ▶ No good methodology for that.
- ▶ In KATAN32 — each key bit difference must enter (at least) two linear operations and two non-linear ones.
- ▶ Hence, an active bit induces probability of  $2^{-2}$ , and cancels four other bits (or probability of  $2^{-4}$  and 6).

# Security Analysis — Related-Key Differentials

- ▶ No good methodology for that.
- ▶ In KATAN32 — each key bit difference must enter (at least) two linear operations and two non-linear ones.
- ▶ Hence, an active bit induces probability of  $2^{-2}$ , and cancels four other bits (or probability of  $2^{-4}$  and 6).
- ▶ So if there are 76 key bits active — there are at least 16 quintuples, each with probability  $2^{-2}$ .

# Security Analysis — Related-Key Differentials

- ▶ No good methodology for that.
- ▶ In KATAN32 — each key bit difference must enter (at least) two linear operations and two non-linear ones.
- ▶ Hence, an active bit induces probability of  $2^{-2}$ , and cancels four other bits (or probability of  $2^{-4}$  and 6).
- ▶ So if there are 76 key bits active — there are at least 16 quintuples, each with probability  $2^{-2}$ .
- ▶ The key expansion is linear, so check minimal hamming weight in the code.
- ▶ Current result: lower bound: 72, upper bound: 84.

# Security Analysis — Related-Key Differentials (cont.)

- ▶ In KATAN48 — each key bit difference must enter (at least) four linear operations and four non-linear ones.
- ▶ Hence, an active bit induces probability of  $2^{-4}$ , and cancels four other bits (or probability of  $2^{-8}$  and 6).
- ▶ Need 61 active bits in the expanded key. We have them.
- ▶ For KATAN64 — need 56.

# Security Analysis — Related-Key Differentials (cont.)

- ▶ In KATAN48 — each key bit difference must enter (at least) four linear operations and four non-linear ones.
- ▶ Hence, an active bit induces probability of  $2^{-4}$ , and cancels four other bits (or probability of  $2^{-8}$  and 6).
- ▶ Need 61 active bits in the expanded key. We have them.
- ▶ For KATAN64 — need 56.
- ▶ Conclusion: no related-key differential in KATAN family.

# Security Analysis — Related-Key Differentials (cont.)

- ▶ In KATAN48 — each key bit difference must enter (at least) four linear operations and four non-linear ones.
- ▶ Hence, an active bit induces probability of  $2^{-4}$ , and cancels four other bits (or probability of  $2^{-8}$  and 6).
- ▶ Need 61 active bits in the expanded key. We have them.
- ▶ For KATAN64 — need 56.
- ▶ Conclusion: no related-key differential in KATAN family.
- ▶ KTANTAN family: still checking computer simulations.

# Questions?

**Thank you for your attention!**