Lightweight Cryptography for IoT

Martin Hell

Crypto and Security Group
Dept. of Electrical and Information Technology
Lund University

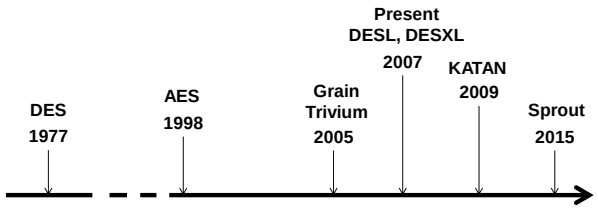
Why lightweight?

- RFID will be used to realize several applications in IoT
- Can be used for e.g., tracking and payment systems
- Passive tags have no battery, still security is often required
  - Eavesdropping, spoofing, cloning
- Even with battery, implementations should be small and power efficient
- Need to develop lightweight cryptographic primitives
- **Goal:** Overview of approaches used to make small (in hardware) and secure encryption algorithms
  - What factors must be considered and how has this been done?
  - Stream ciphers and block ciphers will be covered
  - No details of algorithms, just an overview

What is lightweight?

- Device is resource constrained – attackers are not
  - Lightweight is not same as less secure, but security margin is often traded for lighter implementations
- There is no sharp line defining what is lightweight – goal is to use as little resources as possible
  - Gates (Area)
  - Power consumption
- Speed is still of some importance

Timeline Covered



Hardware cost

- Hardware cost is given as (NAND) gate equivalences
- Rough estimates

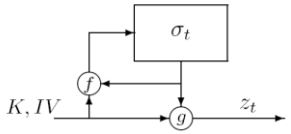
Function	GE
NAND2	1
NAND3	1.5
NAND4	2
XOR2	2.5
D-element	6

- Actual implementation needed for more accurate values

Can we use AES?

- In some places, sure
- Everywhere, no
- AES hardware implementations can be small but then they are quite slow
  - 3100 gates has been demonstrated, using 160 clock cycles [HAHH06]
  - 2400 gates, using 226 clock cycles [MPL+11]
- Quite much work on making AES small

Stream Cipher Principles



$\sigma_0 = \gamma(K, IV)$ **Initialisation function**

$\sigma_{t+1} = f(\sigma_t, K, IV)$ **State update function**

$z_t = g(\sigma_t, K, IV)$ **Output function**

- All these have to be implemented
- Initialisation function reuses the rest – not much cost but very important
  - Robust to protect against chosen-IV attacks
  - Fast in order to have small delay
- 2 functions + 1 state

Bound on Internal State – TMTO [Bab95][Gol97]

- (Known plaintext) Time-Memory Tradeoff on internal state with state of size $N=2^n$
  - Offline: Pick $2^{n/2}$ states and generate keystream, save state-keystream mapping
  - Online: With $2^{n/2}$ keystream, check if mapping exists.
  - State recovery
- **Consequence:** n/2 must be at least size of key, so state must be at least twice the key size

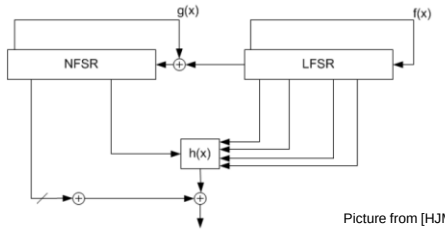
Keysize

Level	Protection	Symmetric	Factoring Modulus	Discrete Logarithm Key	Discrete Logarithm Group	Elliptic Curve	Hash
1	Attacks in "real-time" by individuals *Only acceptable for authentication tag size*	32	-	-	-	-	-
2	Very short-term protection against small organizations *Should not be used for confidentiality in new systems*	64	816	128	816	128	128
3	Short-term protection against medium organizations, medium-term protection against small organizations	72	1008	144	1008	144	144
4	Very short-term protection against agencies, long-term protection against small organizations *Smallest general-purpose level,* *2-key 3DES restricted to 2^{40} plaintext/ciphertexts, protection from 2015 to 2015*	80	1248	160	1248	160	160
5	Legacy standard level *2-key 3DES restricted to 10^6 plaintext/ciphertexts, protection from 2015 to 2020*	96	1776	192	1776	192	192
6	Medium-term protection *3-key 3DES, protection from 2015 to 2030*	112	2432	224	2432	224	224
7	Long-term protection *Generic application-independent recommendation, protection from 2015 to 2040*	128	3248	256	3248	256	256
8	"Foreseeable future" *Good protection against quantum computers, unless Shor's algorithm applies*	256	15424	512	15424	512	512

Table from http://www.keylength.com/en/3/
See [ECRYPT] for ECRYPT II report

Grain [HJM07]

- 80-bit key, 160 bit state
  - Small state
  - Relatively large functions
  - Bit oriented

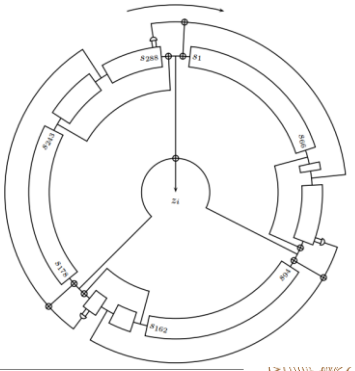


Picture from [HJM07]

Trivium [DP06]

Picture from [DP06]

- 288 bit state, 80 bit key
- Bit oriented
- Very small functions
- Relatively large state



Hardware implementation results

Design	Key bits	Interface bits	Load/Ini cycles	Bits/Cycle (running)	Max. clock freq. MHz	Area NAND GE, gates	Leakage power, µW	Total Power @10MHz, µW
Grain80	80	1	321	1	724.6	1294	2.22	109.45
Grain80, x4	80	4	81	4	694.4	1678	3.24	126.59
Grain80, x8	80	8	41	8	632.9	2191	4.63	150.66
Grain80, x16	80	16	21	16	617.3	3239	7.40	200.53
Trivium	80	1	1333	1	358.4	2599	3.84	181.18
Trivium, x4	80	4	336	4	413.2	2660	4.04	184.83
Trivium, x8	80	8	170	8	359.7	2801	4.45	199.59
Trivium, x16	80	16	87	16	408.2	3185	5.84	231.23
Trivium, x32	80	32	45	32	350.9	3787	7.50	282.55
Trivium, x64	80	64	24	64	348.4	4921	10.68	374.19

Design	Max Throughput, Mbps	Estimated Power, uW	…bit,	…time	…rea, m²	Area-
Grain80	725	7,772				
Grain80, x4	2,778	8,565				
Grain80, x8	5,063	9,247				
Grain80, x16	9,877	11,929				
Trivium	358	6,360				
Trivium, x4	1,653	7,475				
Trivium, x8	2,878	7,024				
Trivium, x16	6,531	9,205	1.41	2.53	395.57	23.3
Trivium, x32	11,228	9,658	0.86	1.75	571.88	16.9
Trivium, x64	22,300	12,677	0.57	1.14	874.14	14.5

For comparison (approx):

RC4: 50000 GE, 10Gbps [SCS+13]
Snow 3G: 11000 GE, 1.72 Gbps [ETSI06]
A5/1: 700 GE [BLM+04]

Figures taken from [GB07]

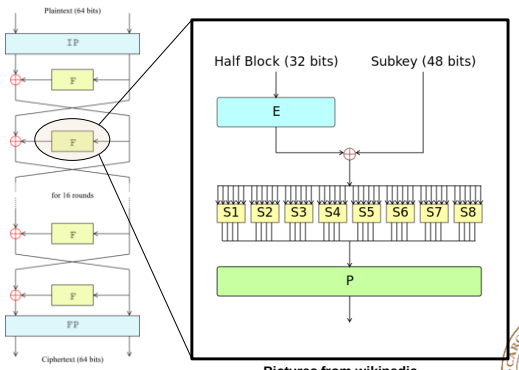
Block ciphers

- Following Grain and Trivium (and many others in 2005), focus shifted to block ciphers
  - Much knowledge about these since e.g., DES and AES
    - Less of a risk to make aggressive designs
- Does not have the same TMTO state bound as stream ciphers
- Bound on memory size
  - Block size – Can be seen as state
  - Key size – Use key to create round keys

DES

- Published 1977, Standardized 1979
  - Differential attack 1990
  - Linear attack 1994
- 56 bit key, 64 bit blocks
- Designed for low hardware footprint from the beginning
  - About 2300 gates if serialized [LPPS07]
    - Area evenly shared by registers, S-boxes and Multiplexors
- **Main problem: Key is too short!**

DES overview



Pictures from wikipedia

Making DES Smaller and Better – DESL and DESXL [LPPS07]

DESL

- Replace the 8 different S-boxes by one S-box
  - Optimized to resist linear and differential cryptanalysis
- Remove IP and FP (=IP^{-1})
  - Save wiring
- 1848 gates

DESXL

- Use key whitening

$$DES_{k,k_1,k_2}(x) = k_2 \oplus DES_k(k_1 \oplus x)$$

- 2168 gates

SP-network

Use several rounds of
[S-boxes -> Permutation layer -> add round key]

Minimizing designs:

- Size/number of S-boxes can be modified
- Permutation layer can be modified
- Key schedule can be modified

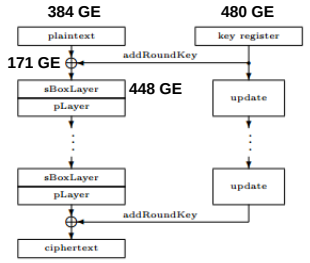
PRESENT [BKL+07]

- 64-bit blocks, **80** or 128 bit key
- 4-bit S-box, 16 times
- Bit permutations
- 31 rounds

```
generateRoundKeys()
for i = 1 to 31 do
  addRoundKey(STATE, K_i)
  sBoxLayer(STATE)
  pLayer(STATE)
end for
addRoundKey(STATE, K_32)
```

Total: 1570 GE

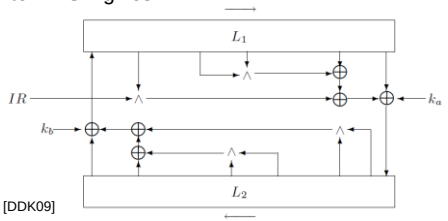
Overview picture from [BKL+07]



KATAN [DDK09]

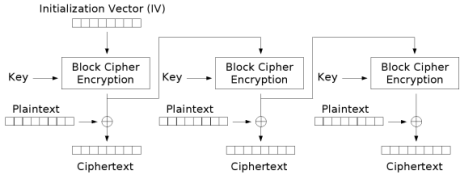
- 32, 48 or 64 bit block size. 80 bit key.
- Based on Trivium, but much smaller
- Key loaded into separate LFSR
- Counter LFSR gives IR

KATAN64: 1054 GE
KATAN48: 927 GE
KATAN32: 802 GE



Picture from [DDK09]

Trivial Attack on Short State – OFB Mode

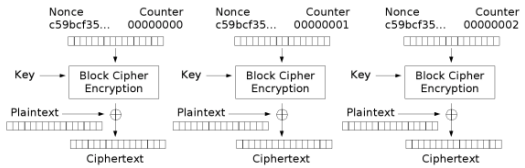


Picture from wikipedia

- We expect keystream period to be 2^{n-1} blocks
- Collision in random sequence after $2^{n/2}$ blocks.

- Distinguishing attack
- See e.g. [EHJ07]

Trivial Attack on Short State – CTR Mode



Counter (CTR) mode encryption

Picture from wikipedia

- Keystream will not repeat since we use a counter as input to block cipher
- Collision in random sequence after $2^{n/2}$ blocks.
- Distinguishing attack
- See e.g. [EHJ07]

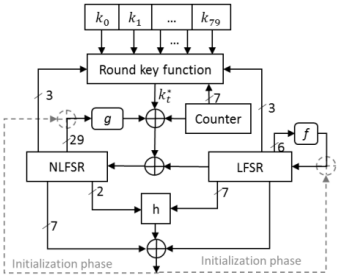
Fixing the key

- Previous designs did not take into consideration that the key needs to be stored somewhere
- In constrained devices the key is typically fixed anyway, so burn it into the device!
- This has been considered in both recent block ciphers and stream ciphers.

Sprout [AM15]

- 40+40 bit state
- 80 bit key burnt into device
- **Idea:** Make key part of the state!
  - TMTO must consider all keys anyway
- 813 gates needed since key is not counted

Picture from [AM15]



Summary and Conclusions

References are to implementations

	GE	Throughput 100KHz (Kbit/s)	Logic
AES [MPL+11]	2400	56.6	0.13µm
DES [LPPS07]	2309	44.4	0.18µm
DESL [LPPS07]	1848	44.4	0.18µm
DESXL [LPPS07]	2168	44.4	0.18µm
PRESENT [BKL+07]	1570	200	0.18µm
KATAN64 [DDK09]	1054	25.1	0.13µm
Grain [GB07]	1294	100	0.13µm
Trivium [GB07]	2599	100	0.13µm
Sprout [AM15]	813	100	0.18µm

- Stream ciphers can be very small and very fast at the same time
- Block ciphers can be based on Feistel networks, SPN or something else – size mostly depends on other properties
- Block ciphers will always have distinguishing attacks that the stream ciphers try to avoid
  - Security comparison is not really fair

References (1/2)

- [AM15] F. Armknecht, V. Mikhalev, "On Lightweight Stream Ciphers with Shorter Internal States*", 2015*
- [Bab95] S. Babbage, "A Space/Time Tradeoff in Exhaustive Search Attacks on Stream Ciphers", 1995
- [BLM+04] L. Batina, J. Lano, N. Mentens, S. B. Örs, B. Preneel, and I. Verbauwhede, "Energy, performance, area versus security trade-offs for stream ciphers", 2004
- [BKL+07] A. Bogdanov, L.R. Knudsen, G. Leander, C. Paar, A. Poschmann, M.J.B. Robshaw, Y. Seurin, and C. Vikkelsoe, "PRESENT: An Ultra-Lightweight Block Cipher", 2007
- [DDK09] C. De Cannière, O. Dunkelman, and M. Knezevic. "KATAN and KTANTAN - A family of small and efficient hardware-oriented block ciphers", 2009
- [DP06] C. DeCannière and B. Preneel. "Trivium specifications", 2006
- [ECRYPT12] ECRYPT – "Yearly Report on Algorithms and Keysizes", 2012
- [EHJ07] H. Englund, M. Hell, T. Johansson, "A note on Distinguishing Attacks", 2007.

References (2/2)

- [ETSI06] ETSI/SAGE, "Specification of the 3GPP Confidentiality and Integrity Algorithms UEA2 & UIA2. Document 5: Design and Evaluation Report", 2006
- [Gol97] J. Golic, "Cryptanalysis of alleged A5 Stream Cipher", 1997.
- [GB07] T. Good and M. Benaissa, "Hardware results for selected stream cipher candidates", 2007
- [HJM07] M. Hell, T. Johansson, and W. Meier. "Grain: a stream cipher for constrained environments", 2007
- [HAHH06] P. Hämäläinen, T. Alho, M. Hännikäinen, and T. D. Hämäläinen, "Design and Implementation of Low-Area and Low-Power AES Encryption Hardware Core", 2006.
- [LPPS07] G. Leander, C. Paar, A. Poschmann and K. Schramm, "New Lightweight DES Variants", 2007
- [MPL+11] A. Moradi, A. Poschmann, S. Ling, C. Paar and H. Wang, "Pushing the Limits: A Very Compact and a Threshold Implementation of AES", 2011.
- [SCS+13] S. Sen Gupta, A. Chattopadhyay, K. Sinha, S. Maitra and B. P. Sinha, "HighPerformance Hardware Implementation for RC4 Stream Cipher", 2013