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


Martin Hell, Lund University
Hardware cost

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

<table>
<thead>
<tr>
<th>Function</th>
<th>GE</th>
</tr>
</thead>
<tbody>
<tr>
<td>NAND2</td>
<td>1</td>
</tr>
<tr>
<td>NAND3</td>
<td>1.5</td>
</tr>
<tr>
<td>NAND4</td>
<td>2</td>
</tr>
<tr>
<td>XOR2</td>
<td>2.5</td>
</tr>
<tr>
<td>D-element</td>
<td>6</td>
</tr>
</tbody>
</table>

- 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

Initialisation function
\[ \sigma_i = \gamma(K, IV) \]

State update function
\[ \sigma_{i+1} = f(\sigma_i, K, IV) \]

Output function
\[ z_i = g(\sigma_i, K, IV) \]

- 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

<table>
<thead>
<tr>
<th>Level</th>
<th>Protection</th>
<th>Symmetric</th>
<th>Factoring</th>
<th>Discrete-Logarithm</th>
<th>Elliptic Curve</th>
<th>Hash</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Attention in &quot;real-time&quot; by individuals. Only acceptable for authentication tag size.</td>
<td>32</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>Very short-term protection against small organizations. Should not be used for confidentiality in new systems. Short-term protection against medium organizations.</td>
<td>64</td>
<td>32</td>
<td>128</td>
<td>32</td>
<td>128</td>
</tr>
<tr>
<td>3</td>
<td>Very short-term protection against small organizations. Short-term protection against medium organizations.</td>
<td>72</td>
<td>1024</td>
<td>144</td>
<td>1024</td>
<td>144</td>
</tr>
<tr>
<td>4</td>
<td>Very short-term protection against small organizations. Long-term protection against small organizations. Medium-term protection against small organizations. Security general purpose level. 2-key, 3DES restricted by 2425 password-hits, protection from 2015 to 2020.</td>
<td>80</td>
<td>128</td>
<td>32</td>
<td>128</td>
<td>32</td>
</tr>
<tr>
<td>5</td>
<td>Very short-term protection against small organizations. Medium-term protection. 3-key, 3DES restricted by 10^17 password-hits, protection from 2015 to 2020.</td>
<td>96</td>
<td>1776</td>
<td>192</td>
<td>1776</td>
<td>192</td>
</tr>
<tr>
<td>6</td>
<td>Long-term protection. Generic application-independent recommendation protection from 2015 to 2040.</td>
<td>112</td>
<td>2432</td>
<td>224</td>
<td>2432</td>
<td>224</td>
</tr>
<tr>
<td>7</td>
<td>Good protection against quantum computers. Unless Shor’s algorithm applies.</td>
<td>128</td>
<td>32</td>
<td>256</td>
<td>32</td>
<td>256</td>
</tr>
<tr>
<td>8</td>
<td></td>
<td>256</td>
<td>512</td>
<td>256</td>
<td>512</td>
<td>512</td>
</tr>
</tbody>
</table>

Table from [http://www.keylength.com/en/3/]

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

---

**Grain [HJM07]**

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

---

**Trivium [DP06]**

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

---

**Hardware implementation results**

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!

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

**DES**

- 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

Overview picture from [BKL’07]

Minimizing designs:
• Size/number of S-boxes can be modified
• Permutation layer can be modified
• Key schedule can be modified

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

Trivial Attack on Short State – OFB Mode

• 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

- 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

Summary and Conclusions

<table>
<thead>
<tr>
<th>References are to implementations</th>
<th>GE</th>
<th>Throughput 100KHz (Kbit/s)</th>
<th>Logic</th>
</tr>
</thead>
<tbody>
<tr>
<td>AES [MPL+11]</td>
<td>2400</td>
<td>56.6</td>
<td>0.13µm</td>
</tr>
<tr>
<td>DES [LPPS07]</td>
<td>2309</td>
<td>44.4</td>
<td>0.18µm</td>
</tr>
<tr>
<td>DESL [LPPS07]</td>
<td>1848</td>
<td>44.4</td>
<td>0.18µm</td>
</tr>
<tr>
<td>DESXL [LPPS07]</td>
<td>2168</td>
<td>44.4</td>
<td>0.18µm</td>
</tr>
<tr>
<td>PRESENT [BK107]</td>
<td>1570</td>
<td>200</td>
<td>0.18µm</td>
</tr>
<tr>
<td>KATAN64 [DDK09]</td>
<td>1054</td>
<td>25.1</td>
<td>0.13µm</td>
</tr>
<tr>
<td>Grain [GB07]</td>
<td>1294</td>
<td>100</td>
<td>0.13µm</td>
</tr>
<tr>
<td>Trivium [GB07]</td>
<td>2599</td>
<td>100</td>
<td>0.13µm</td>
</tr>
<tr>
<td>Sprout [AM15]</td>
<td>813</td>
<td>100</td>
<td>0.18µm</td>
</tr>
</tbody>
</table>

- 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 are to implementations
References (1/2)


References (2/2)