Performance evaluation of software ciphering in UMTS radio network controller

Master’s Thesis, Jukka Jääskeläinen
Nokia Networks

Supervisor: Prof. Timo Korhonen

Agenda

- Objectives of the thesis
- Basic UMTS network architecture
- Confidentiality and integrity protection in the UMTS radio access network
- Performance measurement methods
- Results of the study
- Analysis of the results
- Conclusions
Objectives of the thesis

- The purpose of the study is to find out whether the software implementation of UMTS radio access network encryption is feasible
- Feasibility is evaluated primarily from the performance and capacity point of view

UMTS network architecture (Release 99)

- UMTS system is divided into logical entities
  - Core Network (CN)
  - UMTS Terrestrial Radio Access Network (UTRAN)
  - User Equipment (UE)
- External networks are connected to CN via gateway elements
Radio access network encryption and integrity protection

- Cornerstone is the 128-bit secret key $K$
  - $K$ is a shared secret between USIM smart card in user’s terminal and Authentication Center in user’s home network
  - The keys used in encryption and integrity protection are derived from this key
- Data is transferred encrypted between a terminal and a radio network controller (RNC)
  - In GSM the encryption was terminated already in base station (BS) leaving the potentially vulnerable links between BS and Base Station Controller (BSC) unencrypted
- Encryption and integrity protection are symmetric operations, thus exactly the same algorithm is executed both in terminal and in RNC

Confidentiality algorithm – f8

- f8 is a stream cipher being able to encrypt/decrypt blocks of data between 1 and 20000 bits in length
- Algorithm takes five input parameters and generates random-looking mask that is applied to the plaintext
- Internally f8 uses KASUMI block cipher
Confidentiality algorithm – f8 (cont.)

- KASUMI block cipher is applied as many times as necessary, one KASUMI round produces 64-bit mask
- As a result keystream (KS) is generated

\[
\text{COUNT} \oplus \text{BEARER} \oplus \text{DIRECTION} \oplus 0...0
\]

\[\text{KS}[0]...\text{KS}[63] \quad \text{KS}[64]...\text{KS}[127] \quad \text{KS}[128]...\text{KS}[191]\]

Integrity algorithm – f9

- f9 algorithm is used to implement the integrity protection between a terminal and a network
  - Sending party uses f9 to generate message authentication code (MAC-I)
  - Receiving party uses f9 as well to verify the identity of the sender
- Algorithm takes five input parameters and produces the integrity code that is appended to the end of signaling message
Integrity algorithm – f9 (cont.)

- KASUMI algorithm is also utilized in f9
- The result is 32-bit integrity code MAC-I

![Integrity algorithm diagram]

Performance measurements

- A ciphering software module was implemented for the tests
  - Based on the reference implementation in 3GPP TS 35.202
  - Provides full f8 and KASUMI algorithm functionalities
  - Coded in C, not manually optimized
- An existing hardware-based ciphering implementation serves as a reference
  - Ciphering mask generation (i.e. the f8 algorithm) is done in a separate ASIC circuit
- A test process was also implemented
  - Test process uses both the software ciphering and the hardware ciphering and measures the performance
  - Performance is measured in terms of execution time
  - Average, minimum and maximum execution times are measured
Test environment

- A board with eight Texas Instruments TMS320C55X family DSPs
- The ciphering ASIC connected to DSPs via serial interface
  - ASIC driver process is running in each DSP
- A PC connected to the board via JTAG test interface
  - Used for debugging, program loading, result fetching, etc.

Tests

- Several different kinds of tests were conducted
  - Variable number and size of data blocks to be ciphered
- Most relevant ones map into the data rates and sizes used in real world, i.e. in UMTS
  1. Speech traffic simulation test
     - Data block size is selected to be similar to those used in AMR speech call
  2. Non real-time (NRT) data traffic simulation test
     - Data block sizes are selected to be similar to those in NRT data calls with different data rates
Speech call simulation test

- Measurement results show that software ciphering is significantly faster.
- With three data blocks (same in speech call) the software ciphering consumes about half of the time used by ASIC.
- Difference behaves linearly being about 50% throughout the tested range.

<table>
<thead>
<tr>
<th>no of blocks</th>
<th>ASIC ciph times (us)</th>
<th>SW ciph times (us)</th>
<th>avg diff (us)</th>
<th>avg diff (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>88</td>
<td>37</td>
<td>38</td>
<td>15</td>
</tr>
<tr>
<td>2</td>
<td>88</td>
<td>65</td>
<td>66</td>
<td>31</td>
</tr>
<tr>
<td>3</td>
<td>88</td>
<td>95</td>
<td>96</td>
<td>46</td>
</tr>
<tr>
<td>4</td>
<td>88</td>
<td>126</td>
<td>126</td>
<td>61</td>
</tr>
<tr>
<td>8</td>
<td>88</td>
<td>252</td>
<td>252</td>
<td>123</td>
</tr>
<tr>
<td>12</td>
<td>88</td>
<td>360</td>
<td>364</td>
<td>180</td>
</tr>
<tr>
<td>16</td>
<td>88</td>
<td>502</td>
<td>504</td>
<td>245</td>
</tr>
<tr>
<td>24</td>
<td>88</td>
<td>752</td>
<td>753</td>
<td>367</td>
</tr>
<tr>
<td>48</td>
<td>88</td>
<td>1515</td>
<td>1520</td>
<td>735</td>
</tr>
<tr>
<td>64</td>
<td>88</td>
<td>2027</td>
<td>2027</td>
<td>980</td>
</tr>
</tbody>
</table>

NRT data call simulation test

- Measurement results show that the performance is almost the same with both alternatives.
- With only a few blocks of more than 50 blocks the software is faster, otherwise the ASICs is marginally faster.
- No significant differences.

<table>
<thead>
<tr>
<th>no of blocks</th>
<th>ASIC ciph times (us)</th>
<th>SW ciph times (us)</th>
<th>avg diff (us)</th>
<th>avg diff (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>336</td>
<td>45</td>
<td>46</td>
<td>10</td>
</tr>
<tr>
<td>2</td>
<td>336</td>
<td>78</td>
<td>78</td>
<td>10</td>
</tr>
<tr>
<td>4</td>
<td>336</td>
<td>142</td>
<td>143</td>
<td>159</td>
</tr>
<tr>
<td>8</td>
<td>336</td>
<td>264</td>
<td>266</td>
<td>286</td>
</tr>
<tr>
<td>12</td>
<td>336</td>
<td>425</td>
<td>426</td>
<td>439</td>
</tr>
<tr>
<td>16</td>
<td>336</td>
<td>568</td>
<td>572</td>
<td>576</td>
</tr>
<tr>
<td>24</td>
<td>336</td>
<td>853</td>
<td>861</td>
<td>866</td>
</tr>
<tr>
<td>32</td>
<td>336</td>
<td>1143</td>
<td>1149</td>
<td>1154</td>
</tr>
<tr>
<td>48</td>
<td>336</td>
<td>1720</td>
<td>1722</td>
<td>1736</td>
</tr>
<tr>
<td>64</td>
<td>336</td>
<td>2308</td>
<td>2312</td>
<td>2316</td>
</tr>
<tr>
<td>96</td>
<td>336</td>
<td>3487</td>
<td>3491</td>
<td>3503</td>
</tr>
<tr>
<td>128</td>
<td>336</td>
<td>4678</td>
<td>4680</td>
<td>4692</td>
</tr>
<tr>
<td>196</td>
<td>336</td>
<td>7225</td>
<td>7227</td>
<td>7228</td>
</tr>
<tr>
<td>256</td>
<td>336</td>
<td>9477</td>
<td>9491</td>
<td>9524</td>
</tr>
</tbody>
</table>
NRT data call simulation test (cont.)

Analysis of the results

- According to the results the software ciphering has at least as good performance than the ASIC ciphering
  - Especially when the number of data frames is small or the data frame size is small
- ASIC solution performance suffers from relatively large overhead in inter-process communication and operating system context switches
  - The ASIC solution involves a lot of signaling between the application process and the ASIC driver process
  - The software ciphering does not have any of this overhead because all the processing is done inside the application process

Pros and cons

ASIC pros:
- Already existing solution, tested and integrated

ASIC cons:
- Lower performance due to the interface overhead

SW pros:
- No need for HW design
- Better performance
- Flexible, new functionality can be added later if needed
  - New algorithms etc.
- Rather straightforward to test

SW cons:
- Consumes some of the DSP processing power (max ~8%)
Conclusions

• Software ciphering improves performance, especially for speech traffic ciphering
• It also simplifies the architecture
  • No need for HW-SW interface
  • Faster design cycle
• Implementation is found to be straightforward and require a reasonable amount of time

Thus, the software ciphering is estimated to be a very feasible choice for the purpose.