Secure High-Speed Communication based on Quantum Key Distribution

Norbert Felber
(for Nicolas Gisin)
Overview

- 100Gbit/s Encryption & Quantum Security
- Quantum Key Distribution (QKD) Challenges
- High-Speed Encryption Challenges
- Physically Unclonable Functions
- SHA-3 Hash Functions
Qcrypt + nano-tera.ch Vision

Qcrypt is built on Quantum Information Technologies & is integrating this with Terabit Encryption schemes to improve information security in the 21st century.

ENGINEERING COMPLEX SYSTEMS FOR HEALTH, SECURITY AND THE ENVIRONMENT
QCrypt: 100Gbit/s Encryption & Quantum Security

40..100Gb/s encryption and decryption

highest security for the next-generation communication speed

high-speed Quantum Key Distribution

the impossibility of “Q cloning” is one of the best established facts in Science
QCrypt System: QKD and Fast Encryption

**Alice**
- Key distillation engine
- Random number generator
- CW laser
- Intensity modulator
- Optical attenuator

**Bob**
- Key distillation engine
- Monitoring interferometer

**100 GE**
- TDM
- Optical Link

**10 GE**
- Key Management & Synchronisation

**Key Management & Synchronisation**
- GHASH
- AES

**Quantum channel**

**Annual Plenary Meeting**
April 26th - 27th, 2012
QKD: Rapid Gated Single Photon Detectors

**Challenges**

- Fast pulse amplitude modulation
- Rapid gated single photon detectors
- Stable and high interference visibility
- Synchronization of Alice and Bob and Secret key distillation engine

\[
|\beta_0\rangle := |0\rangle |\alpha\rangle, \quad |\beta_1\rangle := |\alpha\rangle |0\rangle, \quad \langle \beta_0 | \beta_1 \rangle = e^{-|\alpha|^2}
\]
QKD: Rapid Gated Single Photon Detectors

- Peltier cooled InGaAs diode
- Gate rate: 1.25 GHz
- Dead time: 8 ns
- Afterpulse probability: < 1%
- Maximum detection rates: ~100 MHz

Best performance in academia worldwide!
QKD: Secret Key Distillation

**Challenges**

- Fast pulse amplitude modulation
- Rapid gated single photon detectors
- Stable and high interference visibility
- Synchronization of Alice and Bob and Secret key distillation engine

```
|β₀⟩ := |0⟩|α⟩,  |β₁⟩ := |α⟩|0⟩,  ⟨β₀|β₁⟩ = e^{-|α|^2}
```

**Process**

- **Random number generator**
- **Intensity modulator**
- **Optical attenuator**
- **Key distillation engine**
- **Quantum channel**
- **Synchronization and distillation channel**
- **Monitoring interferometer**

**Steps**

- **Sifting**
- **Timing and base information**
- **Error estimation**
- **Random sampling for QBER**
- **Error correction**
- **LDPC forward error correction**
- **Error verification**
- **CRC check**
- **Privacy amplification**
- **Toeplitz hashing**
- **Authentication**
- **Polynomial hashing**
QKD: Secret Key Distillation

- Error correction using LDPC decoder:
  - 802.11n code (Z=81), syndrome encoding
  - **Flexible** code rates ($\frac{1}{2}, \frac{2}{3}, \frac{3}{4}, \frac{5}{6}$)
  - Throughput decrease of 0.5% at 6% QBER

- Privacy amplification using Toeplitz matrices:
  - Matrix-vector multiplication: $10^6 \times 10^5$
  - Random matrix with diagonal structure
  - **Flexible** compression ratio
QKD: Intermediate Results

Results

- Synchronization and sifting channel running
- > 1 Mbps estimated secret key rate up to 19 km
- Stable detection rates over 10 hours (tracking not yet implemented)
Crypt: The Fast Encryption Prototype
Crypt: The Fast Encryption Prototype

100Gbps enCRYPTor
Crypt: Troughput Limits

Challenges
- Alternative 100Gb/s encryption core
- Removing throughput limits
- Strong testbench for network interfaces
- Enhancing system security (PUF)
- Win SHA-3 Hash function contest
Crypt: Troughput Limits

Problem

- Input: 10x10Gb/s gapless packets
  - Encryption produces low overhead (synchronization)
  - Authentication produces high overhead (MAC)
- >100Gb/s data to be transmitted
- Ethernet handles this by discarding packets and resend requests
- Throughput tests with Network Analyzer report <100Gb/s\(^\text{\textsuperscript{\textdagger}}\)
  - idQ: severe commercial disadvantage
\(^\text{\textdagger}}\) depending on packet lengths

Problem solved

- More efficient packet multiplexing method
- Additional link for supplementary data combined from all 10Gb/s channels
- Implemented as 9\(\times\)10Gb/s + 1\(\times\)(10-\(m\))Gb/s, using the \(m\)Gb/s for supplementary data (\(m < 3\) typ.)
- Alternative: separate link (WDM) or service channel of Q-system

\(^\text{\text{*}}}\) currently 3 + 1 for 40Gb/s demo
**Crypt: Troughput Limits**

**Consequences:**
- 88% → 52% ALM usage
- 24% → 21% Block RAMs
- FPGA Place & Route time 20h → 2h

**Changes:**
- adapted
- significantly reduced
- removed
Crypt: System Verilog Test Environment

Challenges
- Alternative 100Gb/s encryption core
- Removing throughput limits
- Strong testbench for network interface debugging
- Enhancing system security (PUF)
- Win SHA-3 Hash function contest
Crypt: System Verilog Testbench

Open Verification Methodology (OVM):
- Complete verification methodology for large gate-count, IP-based SoCs
- Transaction channels: classes providing mechanisms to manage data (FIFO, synchronization, etc.)
- Generation of random values (data, packet lengths)
- Virtual links to access nodes inside the DUT
- Automatic checkers
**Crypt: System Verilog Testbench**

Open Verification Methodology (OVM):

### Example of randomized Ethernet frame

<table>
<thead>
<tr>
<th>Ethernet Frame structure</th>
<th>Preamble</th>
<th>Dest. MAC</th>
<th>Source MAC</th>
<th>Ether type</th>
<th>Payload</th>
<th>CRC/FCS</th>
<th>Inter Frame</th>
</tr>
</thead>
<tbody>
<tr>
<td>Byte number</td>
<td>78</td>
<td>Random</td>
<td>Random</td>
<td>8100</td>
<td>Random</td>
<td>Calculate</td>
<td>1E 1E ... 1E 1E</td>
</tr>
<tr>
<td>Values</td>
<td>5555_5555_5555_D5</td>
<td>Random</td>
<td>Random</td>
<td>2%</td>
<td>100%</td>
<td>100%</td>
<td>0%</td>
</tr>
<tr>
<td>Random values in %</td>
<td>2%</td>
<td>100%</td>
<td>100%</td>
<td>2%</td>
<td>100%</td>
<td>0%</td>
<td>2%</td>
</tr>
<tr>
<td>Size of blocs</td>
<td>Fixed = 8</td>
<td>Fixed = 6</td>
<td>Fixed = 6</td>
<td>Fixed = 2</td>
<td>Random 45 to 1500</td>
<td>Fixed = 4</td>
<td>Random 0(1%), 1(10%), to N</td>
</tr>
</tbody>
</table>

- Generation of random values (data)
- Generation of random packet lengths
QCrypt: System Security?

Crypt: technically secure

Challenges
- Alternative 100Gb/s encryption core
- Removing throughput limits
- Strong testbench for network interfaces
- Enhancing system security
- Win SHA-3 Hash function contest

QKD: provably secure

Side channels need to be secured too!
**QCrypt: System Security?**

**Alice**
- Key distillation engine
- Random number generator
- CW laser
- Intensity modulator
- Optical attenuator

**Bob**
- Key distillation engine
- Monitoring interferometer
- D_{bit}
- D_{mon}

**Key Management & Synchronisation**
- AES
- GHASH
- TDM

**10 GE**
- Optical Link

**100 GE**
- AES
- GHASH

10 GE
- Secret Key

FPCA Firmware update

Replace with box including trojan horse
PUF: Physically Unclonable Functions

Requirements to a good PUF

• is unclonable
• has a large input data width > 48 bits
• provides minimal output data width 80 … 128 bits
• generates random output (for different inputs)

unique physical properties per copy
PUF: Physical Effects

“unique physical properties per copy”
CMOS process variations in VLSI chips

Transistors

- geometry variations
- doping variations
- layer thickness variations
- ....

... capacitance
... leakage currents
... thresholds
...
PUF: Variants

PUFs proposed so far:

- **Race conditions**: delay differences in “equal” pairs of signal paths
- **Ring oscillators**: frequency variations
- **Static RAM (SRAM)**: power-up pattern
- **Optical**: light propagation in passivation layer to on-chip photo diodes

Our proposal:

- **Dynamic RAM PUF** (DRAM PUF)

Patent filed:

“Generating Unique Numbers Using Charge Decay Phenomena”

(the patent covers several other charge-decay based effects suitable for PUFs)
**PUF: DRAM PUF Principle**

- write pattern = PUF input (raw)
- state is stored on capacitors
- refresh is disabled
- leakage (de)charges capacitors
- physical variations
- read word(s) = PUF output (raw)
- sense amplifiers discriminate 0|1
- physical variations
- **observe**: all cells in column contribute to leakage!
**PUF: DRAM PUF Readout Algorithm**

Initialization: find wait time

```
repeat
  • write pattern
  • wait t(k++)
  • read pattern
until 25% cells toggled
```

PUF readout: evaluate function

```
• (input pre-processing)
• write pattern
• wait t_{25%}
• read word(s)
• (output post-processing)
```
# PUF: Comparison

<table>
<thead>
<tr>
<th></th>
<th>unclonable</th>
<th>inputs</th>
<th>outputs</th>
<th>random</th>
</tr>
</thead>
<tbody>
<tr>
<td>Race conditions</td>
<td>yes</td>
<td>some</td>
<td>few</td>
<td>(yes)</td>
</tr>
<tr>
<td>Ring oscillators</td>
<td>yes</td>
<td>some</td>
<td>few</td>
<td>(yes)</td>
</tr>
<tr>
<td>Optical</td>
<td>yes</td>
<td>some</td>
<td>few</td>
<td>(yes)</td>
</tr>
<tr>
<td>SRAM</td>
<td>yes</td>
<td>none</td>
<td>many</td>
<td>no</td>
</tr>
<tr>
<td>DRAM</td>
<td>yes</td>
<td>many</td>
<td>many</td>
<td>yes^</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>repetition rate</th>
<th>cost, effort</th>
<th>platform</th>
</tr>
</thead>
<tbody>
<tr>
<td>Race conditions</td>
<td>high</td>
<td>average</td>
<td>ASIC*, FPGA*</td>
</tr>
<tr>
<td>Ring oscillators</td>
<td>high</td>
<td>average</td>
<td>ASIC*, FPGA*</td>
</tr>
<tr>
<td>Optical</td>
<td>high</td>
<td>very high</td>
<td>opto process*</td>
</tr>
<tr>
<td>SRAM</td>
<td>low</td>
<td>low</td>
<td>ASIC*, FPGA*, Memory*</td>
</tr>
<tr>
<td>DRAM</td>
<td>low</td>
<td>low</td>
<td>ASIC, Memory</td>
</tr>
</tbody>
</table>

* published^ to be proved
PUF: Applications

Some general applications:
• Identification (Challenge – Response)
• Protected encryption key storage
• Pseudo random number generator
• True random number generator (slow)
• En/Decryption (slow)

Some specific applications for QCrypt:
• Q – Crypt boxes pairing
• Protection of Q – Crypt quantum key connection
• Alice – Bob boxes pairing
• Authentication of FPGA bit stream reload (power-up)
• Protection of FPGA firmware update
SHA-3: Hash Functions

HASH Function **BLAKE** by Jean-Philippe Aumasson, Luca Henzen, Willi Meier and Raphael C.-W. Phan: **Final Round of the NIST SHA-3 Candidate Competition** (with Grøstl, JH, Keccak, and Skein)

**Chip SHABZIGER**
- all SHA-3 candidate algorithms
  - ETHZ-IIS:
    - smallest implement. with 2.5Gb/s
  - GMU-CER *):
    - lowest AT product
- Layout of 65nm CMOS Chip (IIS)
- 3 SRAMS for PUF investigations
**Crypt: Intermediate Results**

**Results**

- 40 & 100Gb/s AES GCM core
  - Serpent & OCB replacement
- Optimized network interface with
  - Throughput limit removed
  - FPGA resources freed
  - Strong testbench
- PUF Research started:
  - Patent filed
  - Test chip with PUF SRAMs ready
  - Test chip with PUF DRAMs in fabrication
- SHA-3 candidate BLAKE
  - good chances to win contest against Keccak and Skein

**SHABZIGER:**
- 3 SRAMs for PUFs
- All SHA-3 candidates
- 65nm CMOS, (1.9mm)$^2$

**DYNAMITE:**
- 4 DRAMs 32x64 for PUFs
- cell design, MOS caps
- 180nm CMOS, (1.5mm)$^2$
Left to right: Raphael Houlmann¹, Charles Ci Wen Lim¹, Patrick Trinkler⁵, Olivier Auberson³, Rob Thew¹, Gilles Curchod³, Jeremy Constantin², Pascal Junod³, Nino Walenta¹, Hugo Zbinden¹, Andy Burg², Norbert Felber⁶ (Co-PI), Etienne Messerli³ (Co-PI), Natalia Kulesza⁵, Laurent Monat⁵, Yoan Graf³, Nicolas Gisin¹ (PI), Olivier Guinnard¹, Julien Izui⁴, Christoph Keller⁶, Fabien Vannel⁴
Posters

**UNIVERSITÉ DE GENÈVE**

*A high speed QKD prototype based on the coherent one-way protocol*
based on the coherent one-way protocol
by Nino Walenta, presented by Charles Ci Wen Lim

**Hes∙so**

*100Giga Fast Encryption Engine*

Olivier Auberson, Gilles Curchod, Etienne Messerli

**ETH**

*Physically Unclonable Functions for Secure Hardware*

Christoph Keller

Annual Plenary Meeting  
April 26th - 27th, 2012