GNU Radio for Quantum Optical Communication

Travis S. Humble

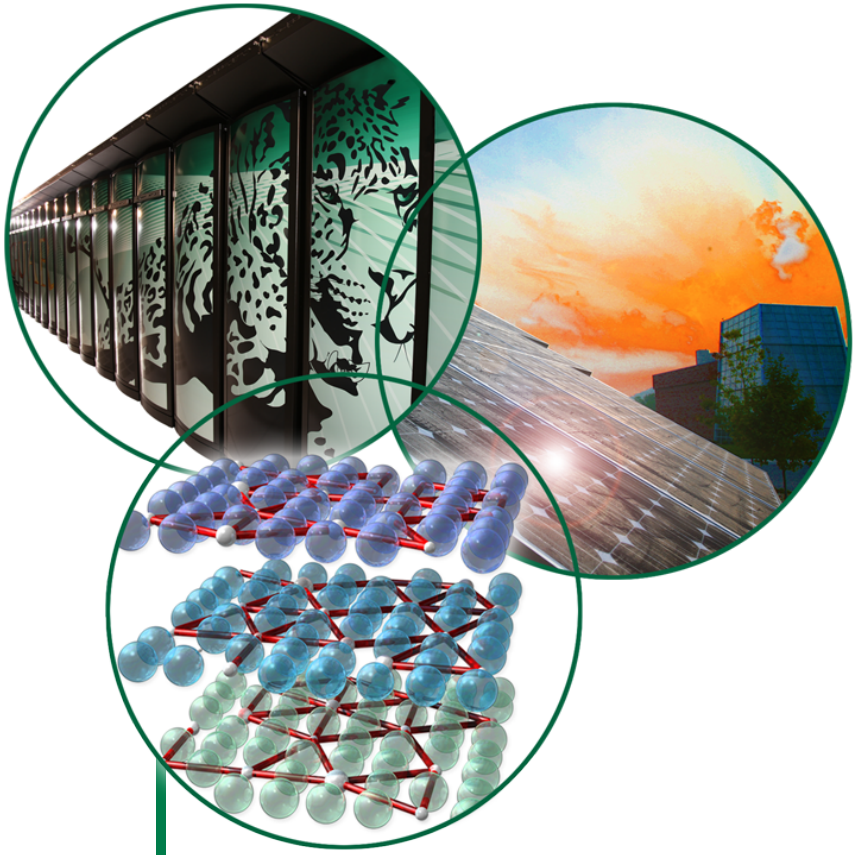
Computer Science and Mathematics

Oak Ridge National Laboratory

In this talk, we discuss the implementation of quantum optical communication protocols within the GNU Radio framework. We report on efforts to develop single-photon optical receivers at infrared and visible wavelengths that interface directly with GNU Radio, and our development of a quantum information tool kit for application testing built on top of GNU Radio.

While the physics underlying quantum communication distinguish it from conventional radios, we conclude that GNU Radio provides a convenient means for prototyping research-grade implementations of various quantum applications and, subsequently, for diagnosing relative performance

This presentation has been authored by a contractor of the U.S. Government under Contract No. DE-AC05-00OR22725. Accordingly, the U.S. Government retains a non-exclusive, royalty-free license to publish or reproduce the published form of this contribution, or allow others to do so, for U.S. Government purposes.



U.S. DEPARTMENT OF ENERGY

GNU Radio Conference
14 Sept 2011 Philadelphia, PA

OAK RIDGE NATIONAL LABORATORY

MANAGED BY UT-BATTELLE FOR THE DEPARTMENT OF ENERGY

GNU Radio for Quantum Optical Communication

- Purpose:
  - Develop a toolkit within the GNU Radio framework for processing quantum communication protocols; prototype quantum applications
  - Develop quantum optical hardware capable of talking with GNU Radio
- Challenges:
  - Interfacing with optical detectors
  - Partitioning quantum communications into GR ‘blocks’
  - Two-way communication
  - Securing the implementation
- Current Status:
  - Prototype optical receivers interface with GR over Ethernet
  - Developed multiple specialty blocks; prototyping simple applications

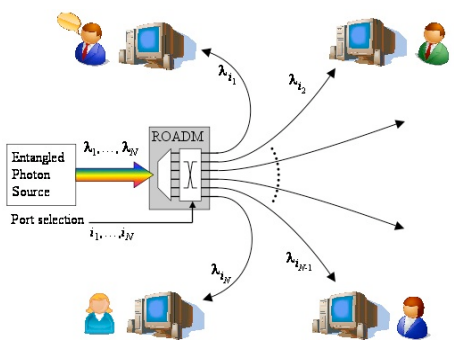
Oak Ridge National Laboratory

Oak Ridge National Laboratory

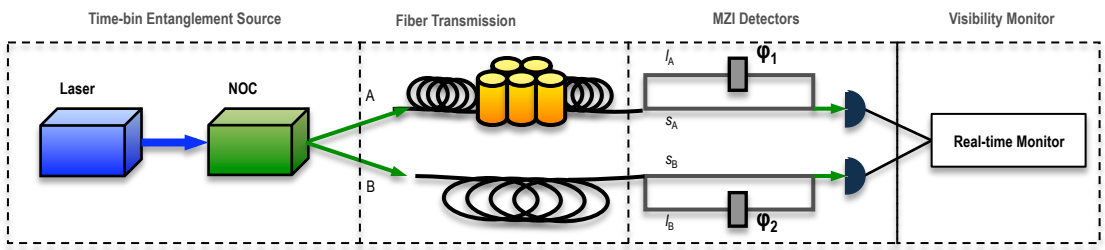
- Department of Energy Research and Development Center
  - Largest DOE open science lab; population 4,800 across 1809 hectares
  - Emphasis in computer science, material science, neutrons, and energy

- Quantum Information Science Working Group
  - Locus of research in quantum information
    - Quantum communication
    - Quantum sensing
    - Quantum computation
    - Quantum imaging

Multi-user Quantum Key Distribution



Quantum Optical Seals



Why GNU Radio?

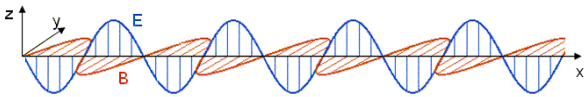
- Leverage advantages of software-defined radio paradigm
  - Replace traditional, dedicated hardware with software-based systems
  - Advantage of being reusable, replicated at reduced cost
  - Advantage of being reconfigurable, tunable, exploratory

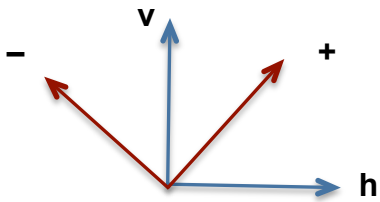
- The advantages for quantum communications
  - Avoid developing research-grade hardware that is quickly obsolete
  - Avoid hardcoding guesses to questions that are not answered yet
  - Rapidly prototype new applications, protocols, and error models
  - Provide a robust research-focused toolkit for enabling discovery

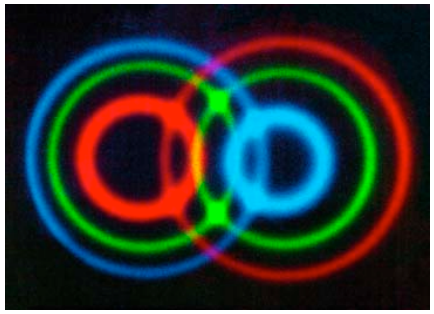
Oak Ridge National Laboratory

Why Quantum Optical Communication?

- Quantum Optical Communication
  - Encoding information into the quantum state of light
    - polarization
    - angular momentum
    - frequency
    - spatial mode
    - temporal mode
    - field quadratures
  - Special features of quantum information
    - No cloning: cannot be copied
    - Non-local correlations: spooky action at a distance
    - Squeezing: sub-shot noise limited detection



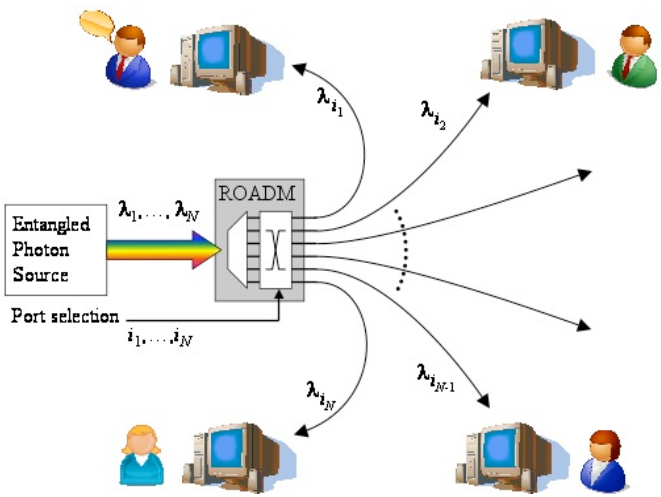


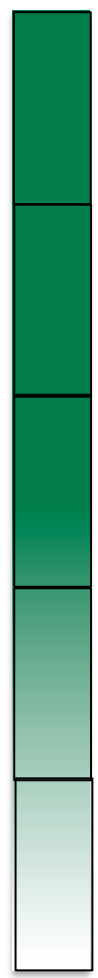


- Promise of Quantum Optical Communication
  - Quantum Key Distribution: Securely distribute cryptographic keys
  - Quantum Teleportation: Relaying quantum information
  - Quantum Optical Seals: Authenticating the integrity of an optical transmission

The Current State of Quantum Optical Communication

- Quantum Communication is relatively young
  - < 20 years of laboratory research
  - Lacks industry standards, but efforts are underway
  - Very few commercial implementations
  - Few commercial users
- Quantum Key Distribution





1911: quantum theory of blackbody radiation

1926: 'photon' invented

1936: quantum logic invented

1957: optical laser invented

1970: conjugate coding invented

1980: entanglement observed

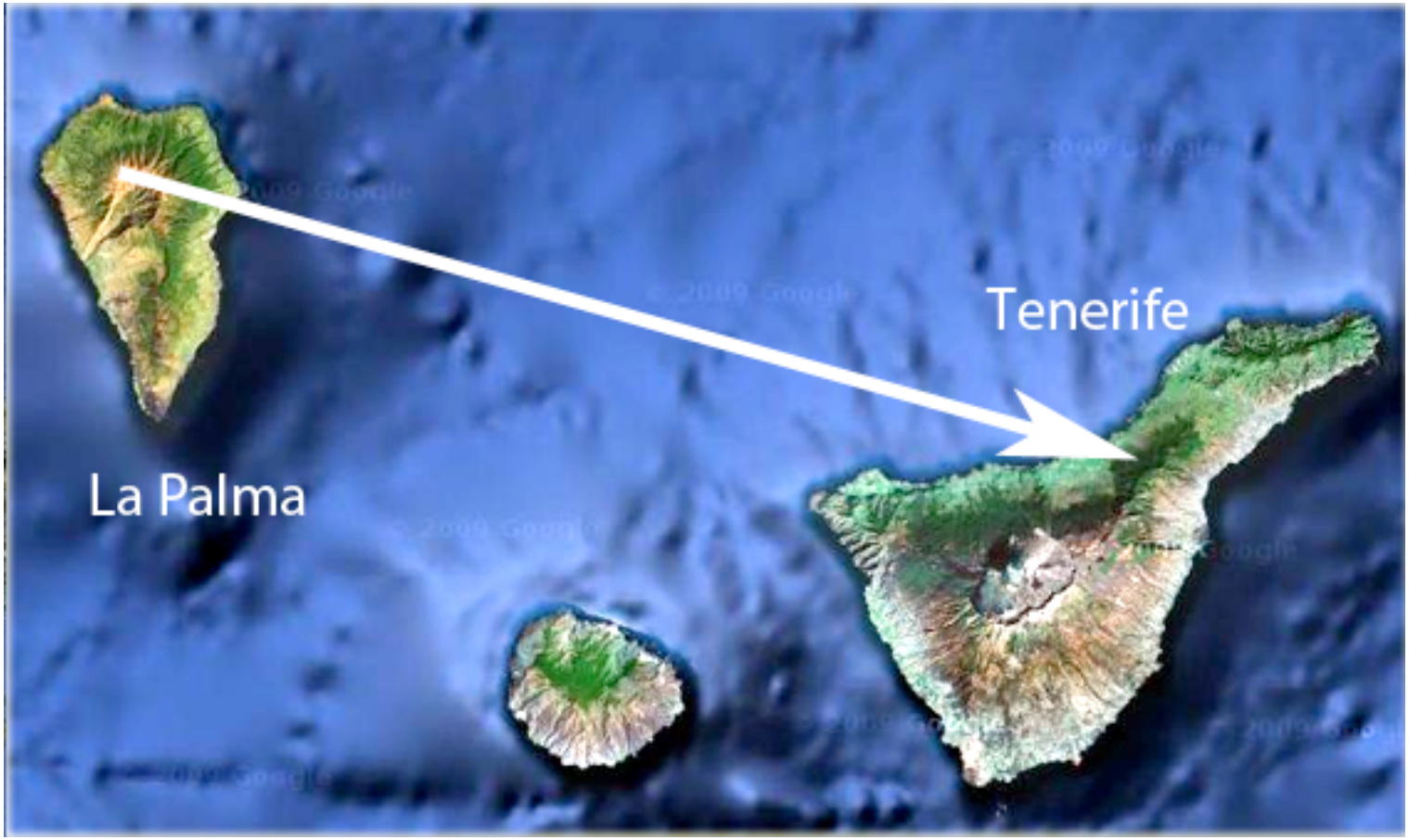
1984: BB84 QKD

1994: Shor's factorization alg.

2003: QKD Network Test beds

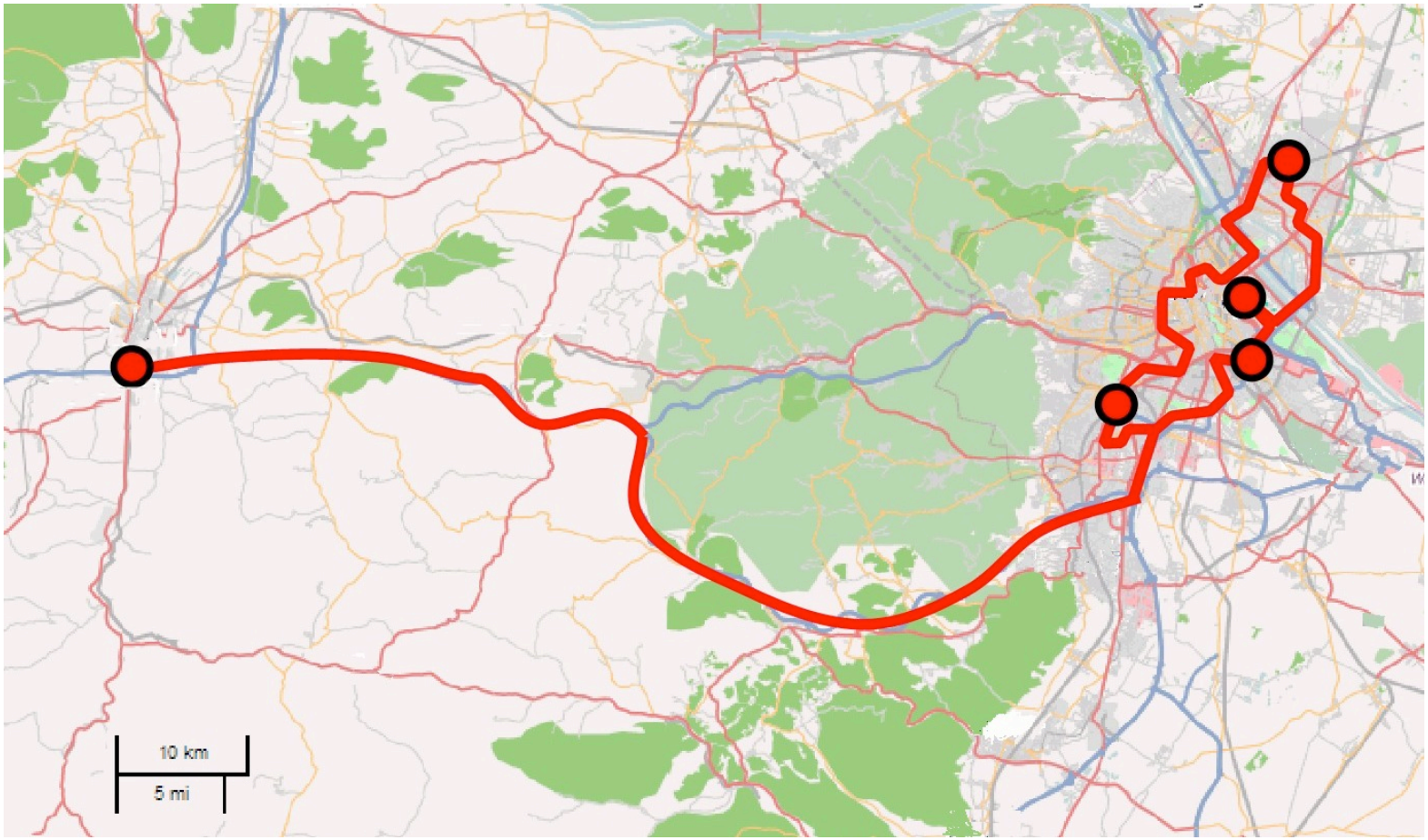
2011

Free-space Optical Channel



Point-to-point free-space QKD over the Canary Islands, 1.8 bit/s @ 144 km by European consortium, 2006-2007

Fiber-based Computer Network



Fiber-ring network in Vienna support QKD base stations, 11kbps @ 20 km by SECOQC consortium, 2004-2010

Fiber-based Commercial System

Quantum Stadium project at 2010 World Cup South Africa,
1kbps @ 25 km by IdQuantique Cerberis

Tokyo QKD Research Network

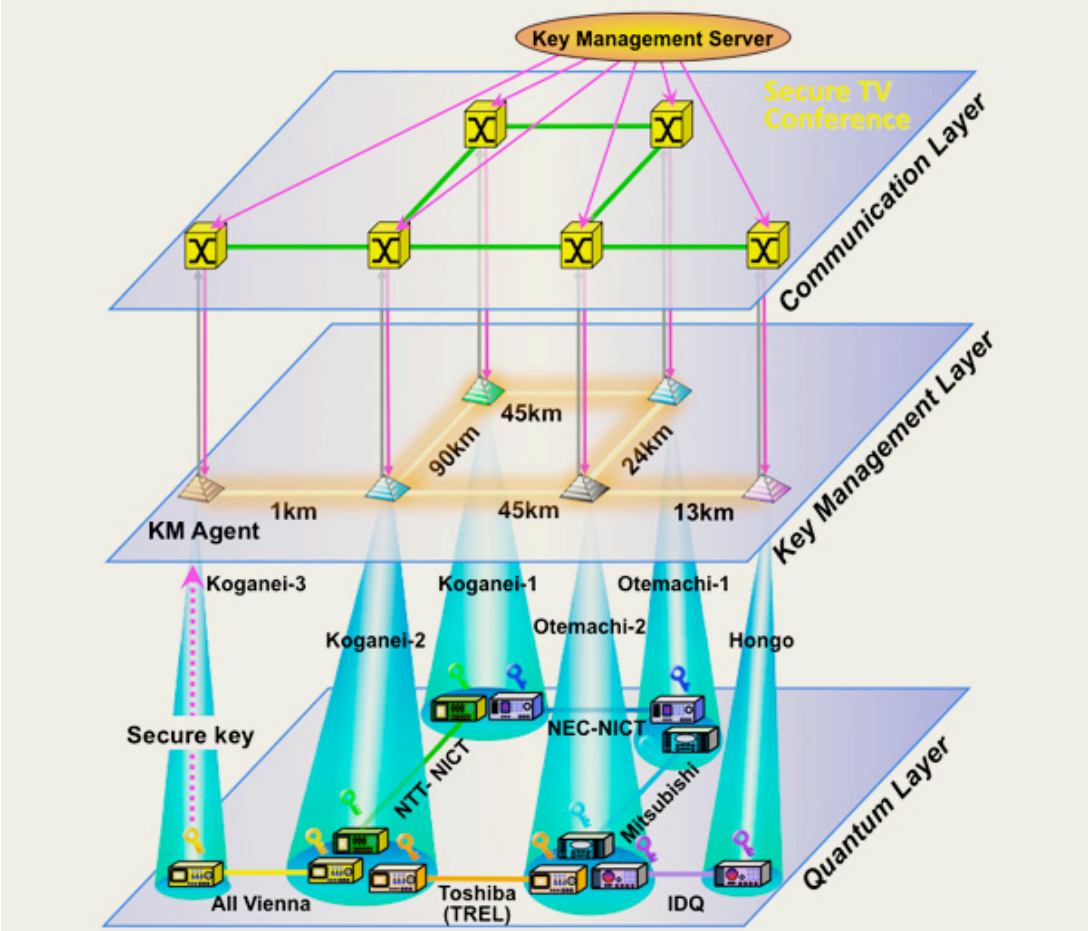
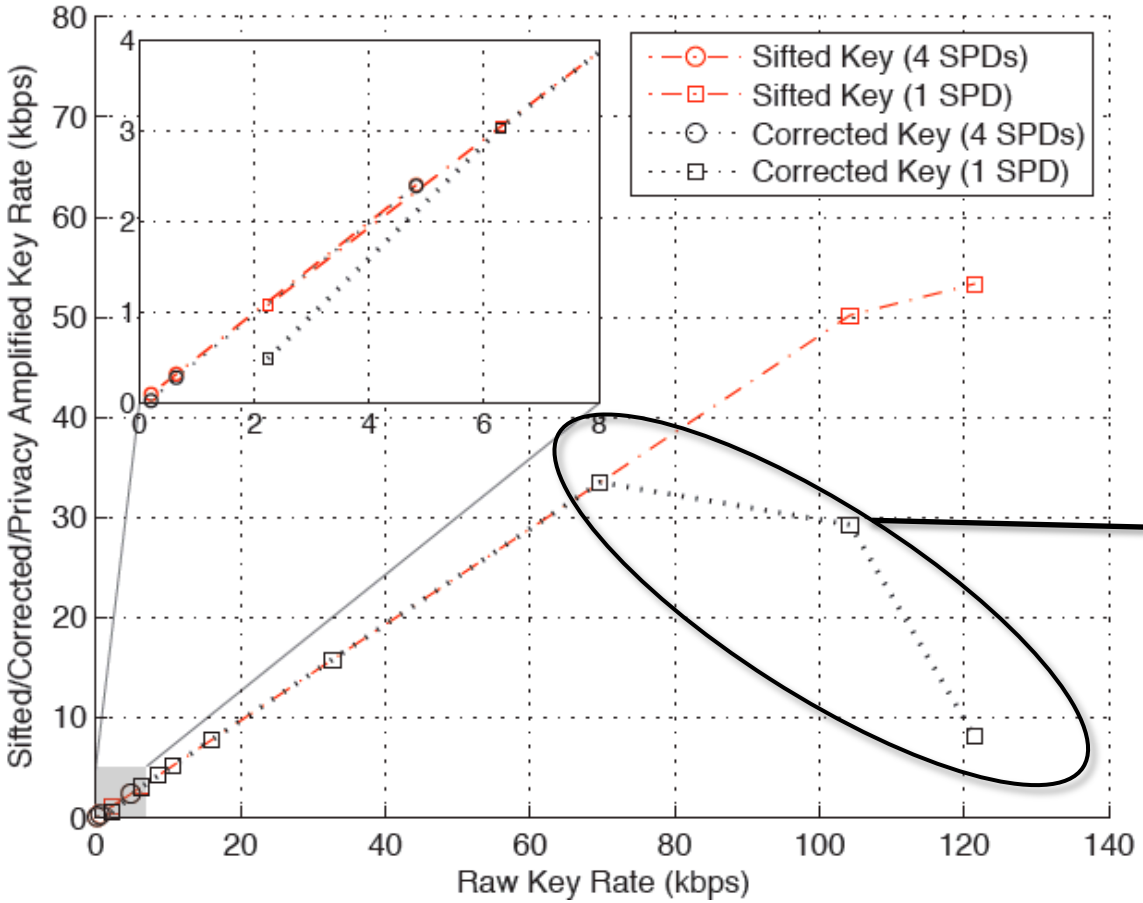


Fig. 2 Network Layer Structure of the Tokyo QKD Network

Metro-area fiber-based network demonstrated by NEC, Mitsubishi, NTT, Toshiba, and others, 40-300 kbps @ 24-90 km

Turn over example

- Figure 3 from “Time-cost analysis of a quantum key distribution system clocked at 100 MHz”, Z. E. Mo et al., Optics Express **19**, 17729-17737 (2011)



Processing can’t keep up with acquisition

Adopting a GNU Radio Paradigm

Hardware Frontend

- Detector converts photons to electrical signals
- Material varies with wavelength

Hardware Backend

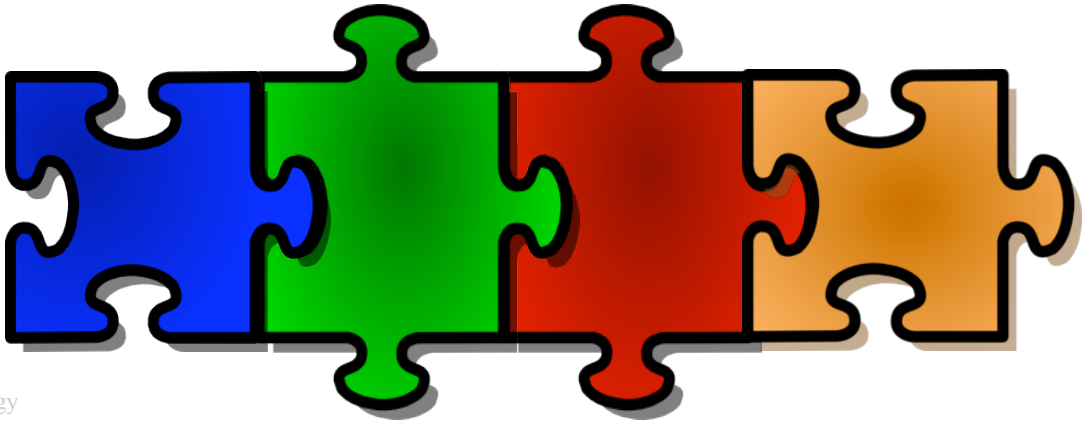
- Process raw electrical signals
- Clocking/synchronization
- Some logic
- Normalizes output

Software backend

- Device driver for hardware backend
- Handle setups, shutdowns, data transfers

Software frontend

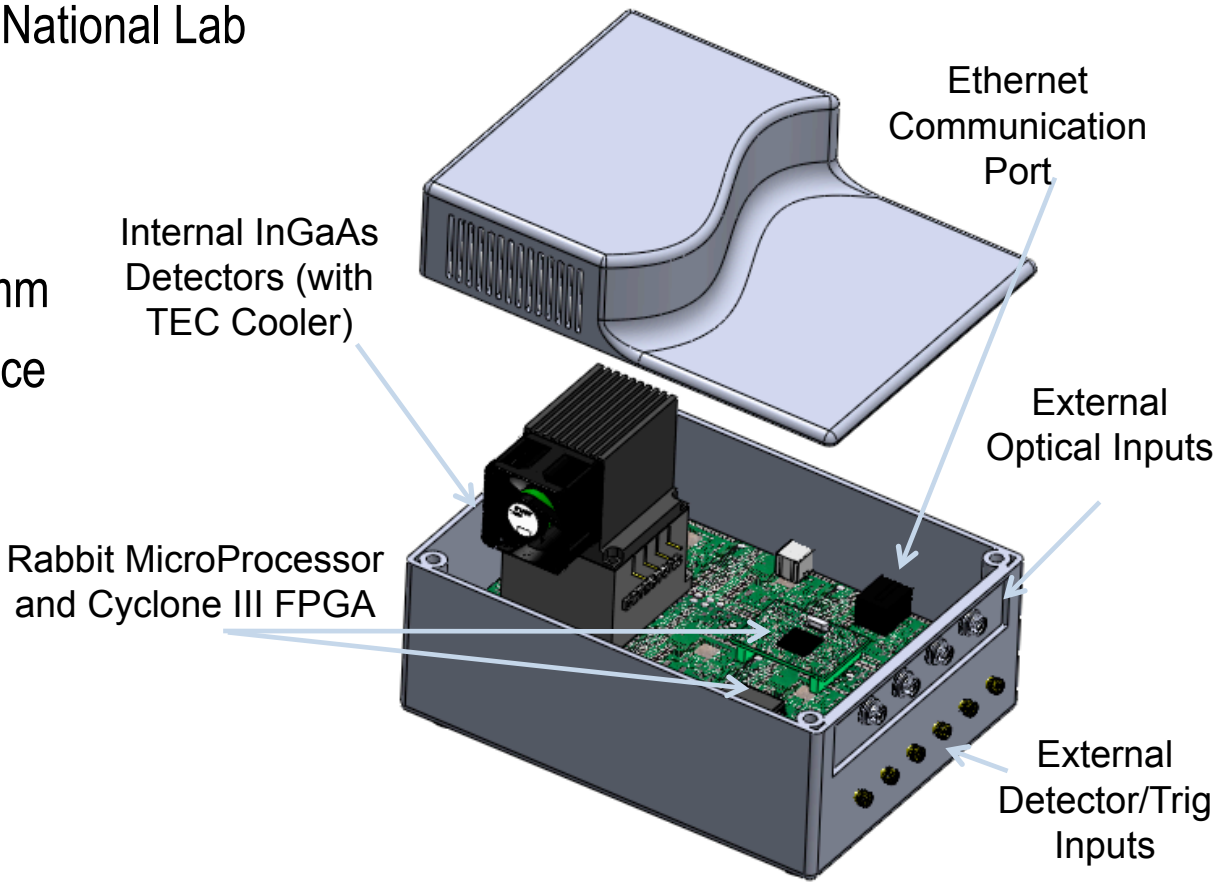
- Software-defined signal processing applications
- Standardized API



Oak Ridge National Laboratory

Quantum Optical Hardware

- ORNL single-photon detectors
- DEED: Duncan Earl's Everything Detector
  - Duncan Earl, Oak Ridge National Lab
- Hardware frontend
  - Single-photon detectors
  - InGaAs: Infrared/Telecomm
  - Si-APD: Visible/Free-space
- Hardware backend
  - Self-differencing circuit
  - Amplifier, comparator
  - Coincidence counting
  - Timestamping
  - FPGA+Microcontroller



Software-defined Quantum Comms

- Quantum Information Tool Kit for Application Testing (QITKAT)
  - GNU Radio-based quantum communication library
  - Developed at ORNL, ongoing (10 Months in)
  - Specialty blocks for handling time-stamped data measured in conjugate bases
  - 64-bit data type: 60-bit counter/metadata, 4-bits for measurements
  - Use _q for quad word (unsigned long long int)
  - List of Blocks from `dir(qitkat)`
    - `timestamp_source_q()`
    - `timestamp_sink_q()`
    - `random_symbols_b()`
    - `entangled_symbols_b()`
    - `fpga_source_q()`
    - `timesync_qq()`
    - `entropy_bf()`
    - `ber_bf()`
    - `bsc_bf()`
    - `visibility_ff()`
    - `detection_ff()`
    - `correct_qq()`
    - `encode_qq()`
    - `sendlist_qq()`
    - `fetchlist_qq()`
    - `coset_qq()`
    - `checkbits_qq()`
    - `discard_qq()`

Oak Ridge National Laboratory

QITKAT Classes

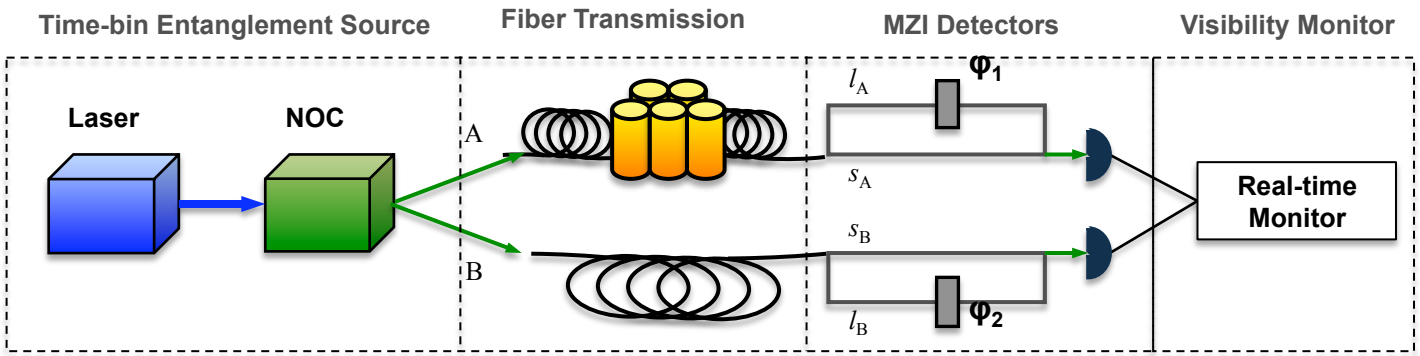
```
// output entries of data as timestamps
timestamp_source_q(uint64_t &data);

// store entries of data as timestamps
timestamp_sink_q(uint64_t &data);

// output uniform random entries of alphabet
random_symbols_b(uint64_t &alphabet, uint32_t seed);

// output uniform random, but correlated entries of alphabet
entangled_symbols_b(uint64_t &alphabet, float correlation, uint32_t seed)

// fetch packetsize data from ipaddr:port using sockets
fpga_source_q(char *ipaddr, char *port, char *packetsize);

// cross-correlate two input streams using sync_length samples every sync_period
timesync_qq(uint sync_length, uint sync_period);

// output either single (joint) entropy of one (two) input stream(s)
entropy_bf(uint32_t num_items);

// output bit error rate between two input streams
ber_bf(uint32_t num_items);

// output result of binary symmetric channel
bsc_bf(float error_rate, uint23_t seed);
```

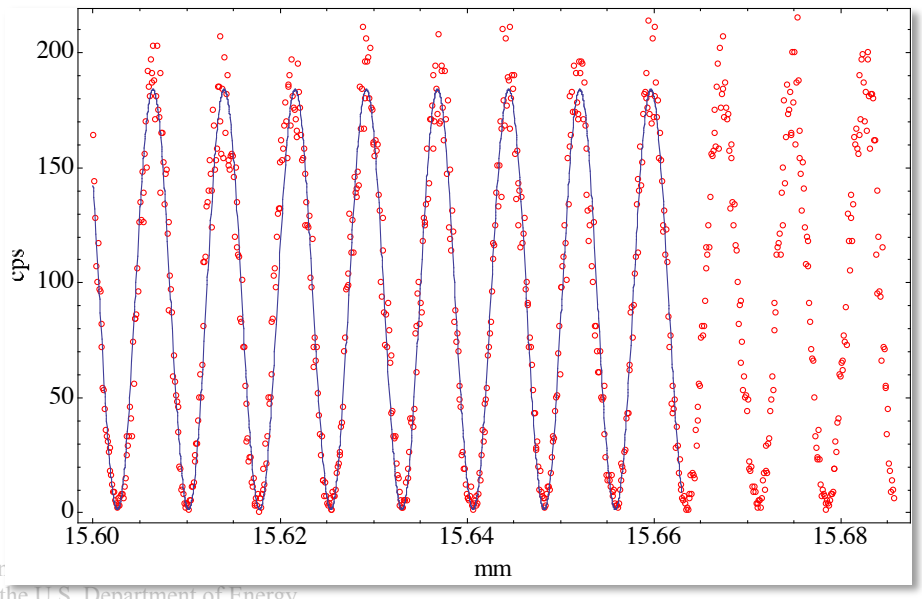
QITKAT Classes

```
// output visibility contrast over window_size samples every step_size samples
visibility_ff(uint32_t window_size, uint32_t step_size);

// output prob. of detection, prob. of false alarm for noise and threshold
detection_ff(float noise, float threshold);

// output error correction using n-by-m generator matrix g
decode_qq(uint32_t n, uint32_t m, uint32_t &g);

// output error encoding using n-by-m generator matrix g
encode_qq(uint32_t n, uint32_t m, uint32_t &g);

// output input after sending list of size element to ipaddr:port
sendlist_qq(char *ipaddr, char *port, uint32_t size);

// output input and list of size elements retrieved from ipaddr:port
fetchlist_qq(char *ipaddr, char *port, uint32_t size);

// output coset of C2 in C1 with input
coset_qq(uint32_t n, uint32_t m, uint32_t &g);

// output unchecked bits after sending errors to ipaddr:port
checkbits_qq(char *ipaddr, char *port);

// output good bits after receiving errors from ipaddr:port
discard_qq(char *ipaddr, char *port)
```

Example: Quantum Optical Seals

- Monitoring integrity of fiber-based transmission of entanglement
  - Absence of non-local correlations indicates intercept-resend attack, intrusion





Presence of entanglement quantified by visibility.

Absence of entanglement signifies an intrusion.

Quantum Seals QITKAT Results

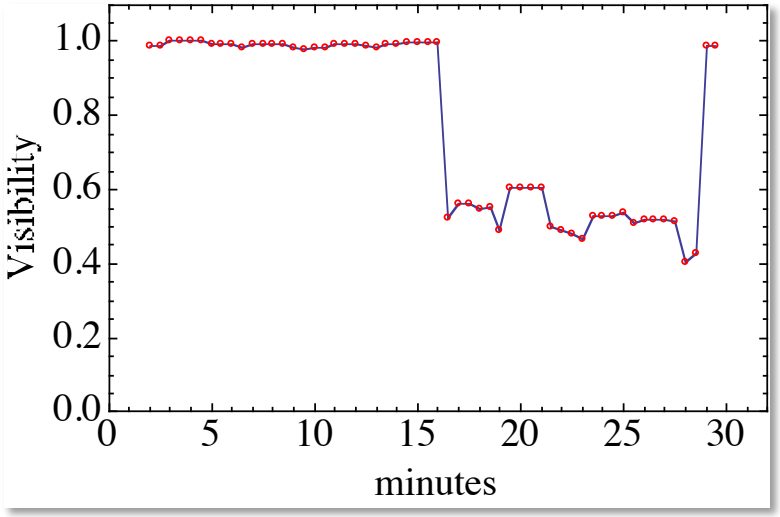
- Entanglement Monitoring: sensing intrusions

fpga_source_q(n) → visibility_qf(n) → detection_ff(n) → alarm_sink_f(n)

visibility.py source code

```
# visibility.py
...
# Source of data
src = gr.vector_source_f(data);

# Compute visibility as (max - min)/(max + min)
vis_window = 120;
vis_step = 30;
vis = qitkat.visibility_ff(vis_window, vis_step);
vissnk = gr.vector_sink_f();

# Compute pdet, pfar
threshold = 1.;
noise = 0.1;
det = qitkat.detection_ff(noise, threshold);

# Sink for running average
snk0 = gr.vector_sink_f();
snk1 = gr.vector_sink_f();
```

Output of visibility.py
running average 120/30

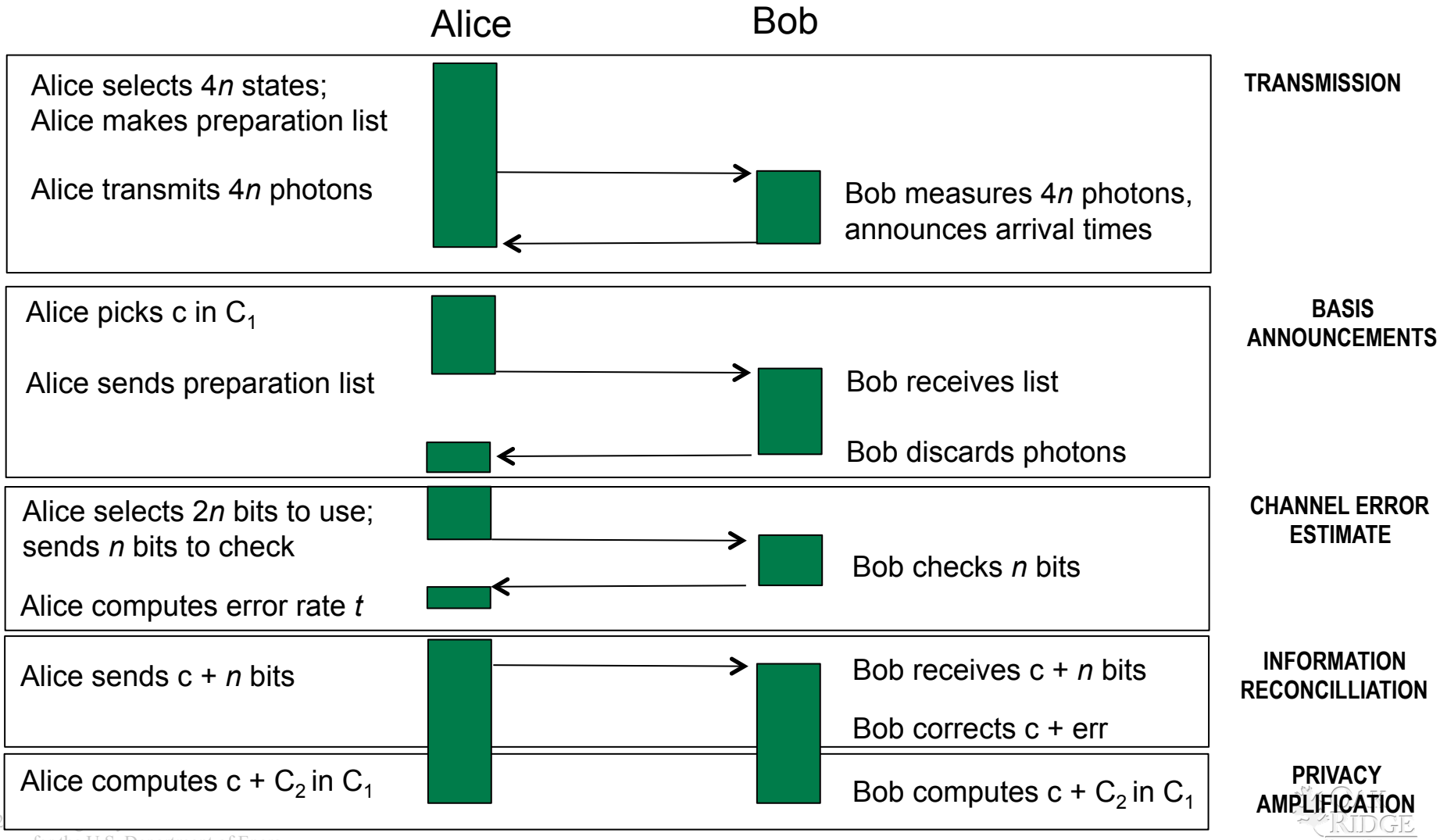


Example: Quantum Key Distribution

- Quantum Communication Protocol:
  - Two separated users, Alice and Bob, securely distribute a one-time pad
  - Consists of 5 distinct phases
    - Transmission
    - Basis Announcements
    - Channel Error Estimate
    - Information Reconciliation
    - Privacy Amplification
  - Many different variants
    - Interaction/communication
    - Processing power
    - Error corrections codes

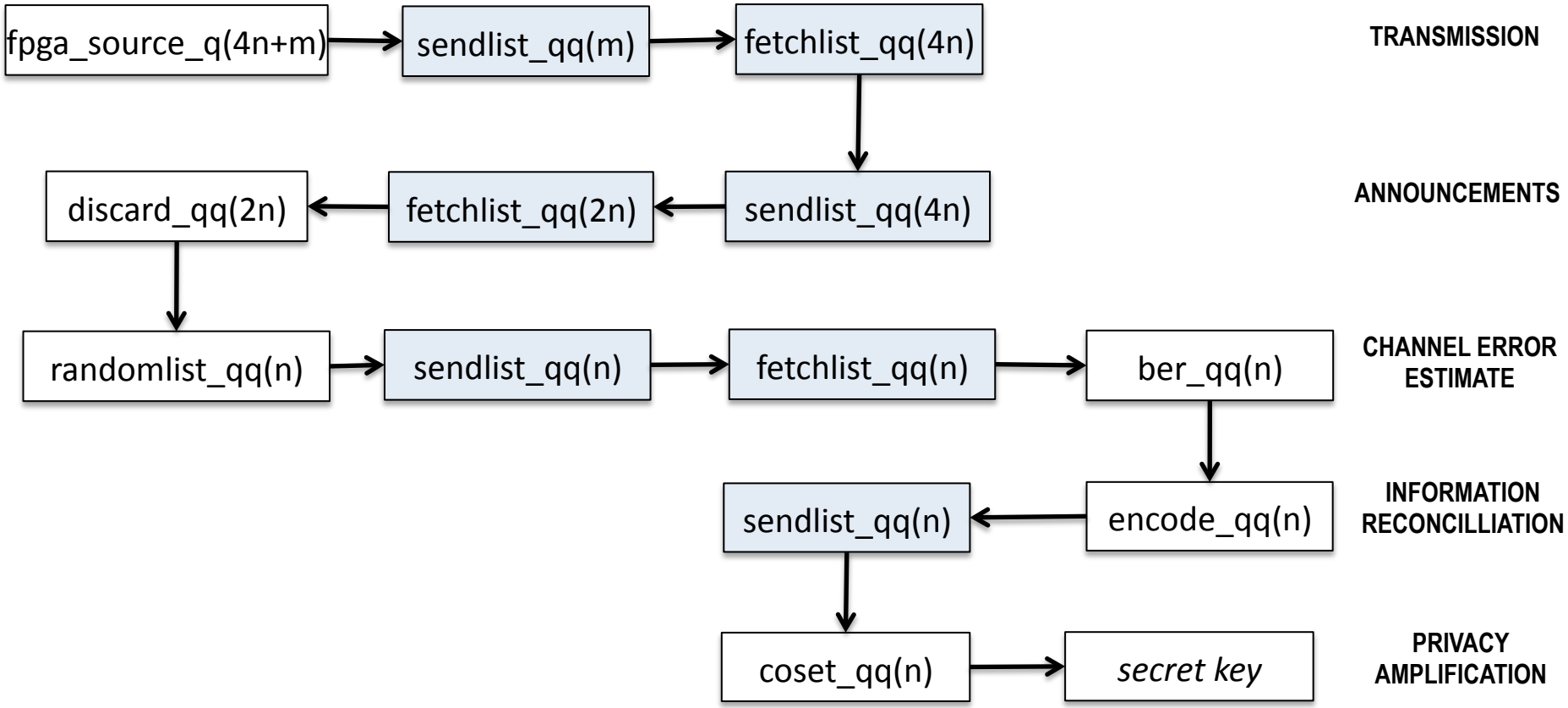
Example: Quantum Key Distribution

- BB84 point-to-point quantum communication (Bennett and Brassard, 1984)



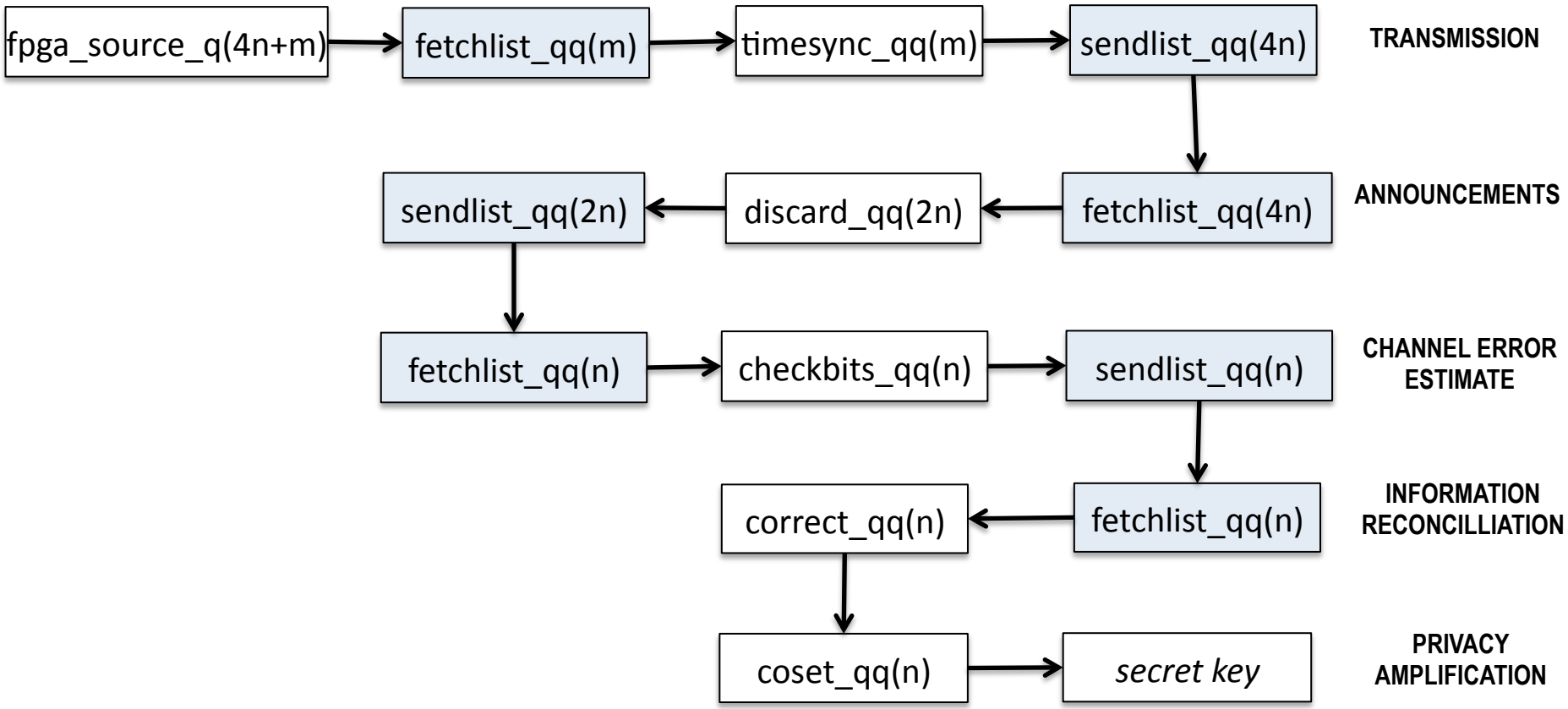
QKD QITKAT Flow Graph

- BB84 Server Side (Alice)



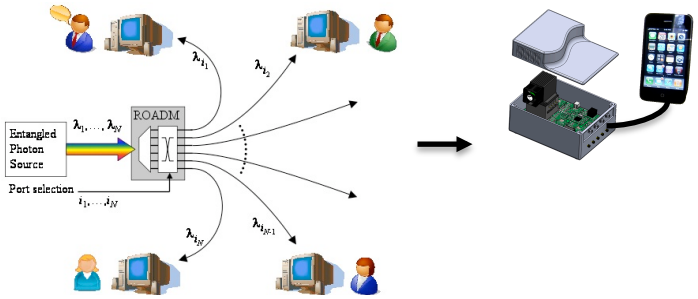
QKD QITKAT Flow Graph

- BB84 Client Side (Bob)



GNU Radio for Quantum Optical Communication

- Summary: Developed QITKAT and DEED's for demonstrating software-defined quantum communication protocols
  - DEED: Single-photon detector integrated with microcontroller
  - QITKAT: GNU Radio-based toolkit for prototyping quantum protocols
  - Quantum Optical Seals
  - Quantum Key Distribution



- The Road Ahead:
  - Demonstrations with DEED's
  - Expansion of QITKAT to other protocols
    - Less interactive QKD protocols

Travis S. Humble
Oak Ridge National Laboratory
865-574-6162
humblets@ornl.gov
www.ornl.gov/~hqt