Automated Control of Quantum Cryptography Schemes

Nathan Greer
Kay Hill
Jeremy Silver
Catherine Thorn

ECE 4006
Summer 2002

July 23, 2002

Georgia Institute of Technology
College of Engineering
School of Electrical and Computer Engineering
1. Abstract

The purpose of this project was to design and implement a system for automated control of a quantum cryptographic system. This task had four parts: 1) to generate a 1 MHz clock signal and synchronized random bits to control the data sent by the transmitter, 2) to convert the random TTL signals into usable currents, 3) to convert the output of the receiver into TTL signals of sufficient duration, and 4) to read and store these TTL signals and perform analysis on the data. The short 10-week schedule and limited manpower forced the team to focus on completion of the first three tasks. Software was used to generate a 1.0 MHz clock signal and synchronized random TTL outputs. These outputs were converted by means of a simple voltage divider/transistor circuit. To convert the 5 ns, 200 mV receiver signal to TTL, a high-speed comparator was cascaded with a monostable multivibrator. The output was then clocked into a shift register/buffer system and read by a National Instruments I/O card. The team was able to complete the design portion of this project, however, implementation was limited by several factors. On the sending side, the I/O card exhibited unexpected duty cycle problems during transmission of the clock signal. The timing characteristics of the receiver took weeks to accurately define and were then too fast to accommodate even high-speed TTL components. Despite these setbacks, the sending side voltage conversion and the receiving side buffer system were successfully implemented.
2. Introduction

This project was a collaborative venture between undergraduate electrical and computer engineering students from the Georgia Institute of Technology in Atlanta, Georgia, and students completing graduate work at Georgia Tech Lorraine in Metz, France. As the first collaboration of this kind between the two Georgia Tech campuses, it also served to test the feasibility of a senior design experience abroad. As the undergraduate contributors, our responsibility was to design and implement a system for automated control of an already functional quantum cryptography scheme. This involved designing transmitter and receiver interfaces to a) control the data being sent and b) store the received data for later analysis and reconciliation. These interfaces will be crucial in the final implementation of quantum key distribution, an on-going Georgia Tech Lorraine (GTL) - Centre National de la Recherche Scientifique (CNRS) Telecom research area.

3. Background

3.1 Quantum Cryptography

For messages to maintain secrecy, it is common for encryption to be performed in a reversible manner. A cryptographic key is used to encode the message in a way that the message can only be decoded with knowledge of the original key. When both transmitter and receiver possess this key and its secrecy is maintained, they can communicate in complete privacy. Code complexity has increased as technology has advanced, but code-breaking methods have likewise improved and there is an increasing demand for unbreakable keys.

Quantum cryptography (QC) is a method by which messages can be sent from transmitter to receiver while maintaining strict confidentiality. This technique revolves around the laws of
quantum mechanics and the use of a new, random key for each transmission. In order to maintain secrecy, the key must be at least as long as the message. Theoretically, the transmitted key is absolutely unbreakable, although practical applications may not allow for perfect security.

There are a limited number of researchers who have successfully performed quantum cryptography and most use a similar strategy. The procedure involves the transmission of a photon through optical fibers. In this experiment, a laser diode is used to produce a single photon, which is then attenuated. A random bit generator is used to control the phase (0, π/2, π, or 3π/2 radians) of a voltage-controlled oscillator operating at a fixed frequency, but with a varying phase. This random phase is the input to the phase modulator that transmits the individual photon over optical fibers to the receiver, where an identical random phase modulation system controls a filter that precedes a Fabry-Perot Interferometer (used as a spectrum analyzer) that then leads to a photon counter [1].

The first protocol for QC, the BB84 protocol, remains one of the most popular methods for obtaining a secure encryption key. Proposed in 1984 by Charles H. Bennett, from IBM New York, and Gilles Brassard, from the University of Montreal, it requires that each of four photon orientations be described by a series of two bits, as shown in Table 1. The first bit, referred to as the photon’s “base,” describes whether the orientation is diagonal or rectilinear. A “1” specifies that the photon is diagonal, either π/2 or 3π/2 radians; if the base is “0,” the transmitted bit is rectilinear, either 0 or π radians. The second bit, simply referred to as the photon’s “bit,” defines the two remaining possible orientations of the photon. For instance, horizontally oriented photons could be described with a “0” bit while a “1” bit describes vertical orientation. Similarly, a π/2 radian oriented photon could have a “0” bit, while a 3π/2 radian oriented photon could have a “1” bit. The four orientations of a photon can then be specified with only two
binary numbers. For example, a horizontal photon would be identified with the base and bit series “0, 0” while a $\pi/2$ radian photon would be identified with the series “1, 0” [2]. For reconciliation purposes, it is important that the transmitter, commonly referred to as Alice, maintain a copy of the bit sequence she sends.

<table>
<thead>
<tr>
<th>Photon Orientation</th>
<th>Base</th>
<th>Bit</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\rightarrow$ (0)</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$\uparrow$ ($\pi$)</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>$\nearrow$ ($\pi/2$)</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>$\swarrow$ (3$\pi/2$)</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

The receiver uses polarized filters to sense when a photon has been transmitted. It is crucial that the receiver, commonly referred to as Bob, use a different, randomly oriented (rectilinear or diagonal) filter for every possible reception period. When Bob chooses a rectilinear filter, he will theoretically detect a vertically oriented photon (bit = 1) 100% of the time, a horizontally oriented photon (bit = 0) 0% of the time, and a $\pi/2$ or $3\pi/2$ oriented photon 50% of the time. Thus, if Bob chooses the same base as Alice, he knows with 100% certainty which bit she sent. Bob then communicates with Alice over a public channel the filter orientation he chose, but not whether or not a photon was detected. Alice then communicates over the public channel whether or not Bob chose the filter orientation that would allow him to correctly detect the transmitted photons. Alice and Bob discard the bits transmitted during incorrect filter orientations thereby obtaining a secure encryption key [3]. Table 2 on the next page demonstrates this logic.
Table 2. Logic used to obtain a secure encryption key, adapted from [3].

<table>
<thead>
<tr>
<th>Alice’s random transmission</th>
<th>( (0,1) )</th>
<th>( (1,0) )</th>
<th>( (0,1) )</th>
<th>( (0,0) )</th>
<th>( (1,1) )</th>
<th>( (0,1) )</th>
<th>( (1,0) )</th>
<th>( (1,0) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bob’s random receiving base</td>
<td>R (0)</td>
<td>D (1)</td>
<td>D (1)</td>
<td>R (0)</td>
<td>R (0)</td>
<td>D (1)</td>
<td>D (1)</td>
<td>R (0)</td>
</tr>
<tr>
<td>Bits received</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Communicated Base</td>
<td>R (0)</td>
<td>D (1)</td>
<td>R (0)</td>
<td>R (0)</td>
<td>D (1)</td>
<td>D (1)</td>
<td>R (0)</td>
<td>D (1)</td>
</tr>
<tr>
<td>Alice checks for correct base</td>
<td>OK</td>
<td>OK</td>
<td>OK</td>
<td>OK</td>
<td>OK</td>
<td>OK</td>
<td>OK</td>
<td>OK</td>
</tr>
<tr>
<td>Secret Bits</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Secret Key</td>
<td>( (0,1) )</td>
<td>( (1,0) )</td>
<td>( (0,0) )</td>
<td>( (0,0) )</td>
<td>( (1,0) )</td>
<td>( (1,0) )</td>
<td>( (1,0) )</td>
<td>( (1,0) )</td>
</tr>
</tbody>
</table>

The experiments currently being performed by GTL/CNRS Telecom are performed with only 1 bit of information for each photon, which calls for a slightly modified protocol. Using this method, Bob does not communicate with Alice the filter orientation he used, but whether or not he detected a photon. Alice and Bob then establish an encryption key by dropping the undetected bits.

The BB84 protocol, along with several other established QC protocols such as the 2-state and 6-state protocols, and the modified protocol used by the GTL/CNRS lab, is characterized by specific error statistics. The probability that Bob will detect a photon is relatively stable, as is the probability that he will have chosen the correct bit or filter orientation. If an eavesdropper, commonly referred to as Eve, attempts to listen to the photon transmission from Alice to Bob, it is theoretically impossible for her presence to go unnoticed because when Eve attempts to gain information from a transmitted photon, she alters the state of that photon. During reconciliation, Alice and Bob would detect the presence of Eve because of the change in their error statistics [4]. Ongoing research at GTL will examine how Eve can be detected and other effects of her presence on the QC system.
3.2 *Important Protocol Components*

3.2.1 *Bi-phase shift key modulators*

Bi-phase shift key (BPSK) modulators are used to control the orientation of the transmitted photon. Depending on the input current (±20 mA), the BPSK phase shifts a waveform of fixed frequency by 0° or 180°.

3.2.2 *Avalanche photo diodes*

Bob uses an avalanche photo diode to detect the presence of the transmitted photon. The diode outputs a 2-5 ns, 200 mV “avalanche” when a photon is detected.

4. *Objectives*

4.1 *Project Definition*

The purpose of this project was to design and implement the computer control of an existing quantum cryptography scheme. The specific tasks are outlined below:

a. Using software, generate random bits to control the BPSK of the sender, Alice. In addition, generate a clock signal of up to 1 MHz.

b. Using hardware, convert the TTL computer output into an appropriate BPSK input, i.e. ±20 mA.

c. Using hardware, convert the photo diode output into TTL signals to be inputted into the computer.

d. Using software, record information regarding photon detection and reconcile the transmission to obtain an encryption key.
4.2 **Design Constraints**

The GTL laboratory provided a programmable National Instruments PCI-6534 I/O card which operated at TTL voltage levels. The input into the BPSKs had to be variable up to \( \pm 20 \) mA. All components had to function at clock speeds up to 1 MHz and be compatible with both the current QC protocol used in the lab as well as the complete BB84 protocol.

4.3 **Challenges and Anticipated Problems**

We expected to encounter several technical issues in completing this project, including those associated with synchronization, modularity, and timing. On the transmitting side, synchronizing the clock signal with the randomly generated bits was required for accurate functioning of the system and reconciliation. Similarly, on the receiving end, alignment of the demodulated clock signal with the avalanche-detect output was imperative for correct counting and storage of data. Interfacing with the avalanche photo diode was expected to be very difficult due to the extremely short duration and low amplitude of the output pulse. Operating at a clock speed of up to 1 MHz would provide an additional challenge.

We also anticipated many non-technical issues. This being the first collaborative project of its kind between graduates and undergraduates, French and American, we expected some communication and cultural difficulties, especially regarding language and work schedule differences. The unfamiliar environment and limited lab resources would also provide significant challenges. Additionally, none of the team members were familiar with the equipment provided, or traditional implementations of the project tasks, making our ten-week schedule very ambitious.
4.4  Project Planning and Logistics

4.4.1  Task Groups and Division of Labor

In order to most efficiently accomplish our goals, we decided to divide into two groups: software and hardware. Jeremy Silver and Nathan Greer undertook the software components of this project while Catherine Thorn and Kay Hill were in charge of the hardware components.

4.4.2  Project Timeline Comparison

Our projected and actual project timeline are shown below in Table 3.

<table>
<thead>
<tr>
<th>Week</th>
<th>Projected Schedule</th>
<th>Actual Schedule</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Specifications</td>
<td>Specifications</td>
</tr>
<tr>
<td>2</td>
<td>Main Design</td>
<td>Main Design</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Purchase and</td>
<td>Purchase,</td>
</tr>
<tr>
<td></td>
<td>Implementation</td>
<td>Implementation,</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>and Modification</td>
</tr>
<tr>
<td>6</td>
<td>Design and</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Modification</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Testing</td>
<td>Testing</td>
</tr>
<tr>
<td>9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Report</td>
<td>Report</td>
</tr>
</tbody>
</table>

5.  Transmitter

5.1  Design

Control of Alice’s phase shifters was implemented by using software to generate a 1.0 MHz clock signal as well as TTL random base and bit signals. These signals were then converted into the ±20 mA input to the BPSK. The complete block diagram is shown in Figure 1.
To generate the TTL signals, the software, which can be found in Appendix A, combined bit patterns from two arrays (base and bit) with an alternating 0/1 pattern in order to generate the three output signals. The two arrays were loaded with both random bit patterns and known bit patterns so that the same "random" data could be sent multiple times during experiments. To generate the random patterns required, the last two bits of both the base and bit arrays were XORed, generating a new bit that was fed back into the arrays themselves, as well as used in the final output bit streams. In order to alleviate the bottleneck at the PCI bus of the NI-DAQ card, which is shared by several resources, the data for the clock, base and bit signals was loaded into the I/O buffer of the card. The data loaded into the buffers was responsible for generating the required 1.0 MHz clock signal, in addition to controlling the bit and base. Once the data was loaded into the buffers, the streams of data were outputted through three of the pins on a single output port of the card. The outputs for the photon bases and bits were connected to a voltage conversion circuit in order to obtain the ±20 mA required to control the BPSK.

The circuit shown in Figure 2 accomplishes voltage conversion by means of a resistor divider network which produces +1.66 V when the input is 5.0 V and -1.66 V when the input is 0.0 V. The positive voltage biases the npn transistor, causing an approximately 0.65 V drop.

![Figure 1. Block diagram of Alice-side control circuitry.](image)
across the transistor, and producing an output voltage of +1.0 V. Similarly, the negative voltage produces a –1.0 V output. This output is converted into the proper current input by means of a 2.0 kΩ variable resistor.

Figure 2. Hardware implementation of voltage converter.

5.2 Testing:

The software portion was tested by loading a known 3-bit pattern into the base and bit arrays (010). Loading the arrays with a 3-bit pattern produced seven clock cycles worth of data as the result of XORing the bits (see Appendix B). The software was tested using 3-bit arrays because this allowed the entire pattern to be easily observed with an oscilloscope.

The circuit shown in Figure 2 was assembled on a protoboard and tested using a function generator to simulate TTL input. The output behaved as expected, producing +1.0 V when the input was high and –1.0 V when the input was low. The circuit was then connected to the BPSK and used to control the phase of the output. A high input produced an output that was 180 degrees out of phase; a low input produced an in-phase output.
5.3 Results:

The output of the clock data stream was connected to a scope where a square clock pulse was measured, having a frequency of 1.1 MHz. While the initial analysis of the generated clock signal appeared valid, further analysis showed that the duty cycles for the rising and falling clock edges were not symmetric; the rising edge comprised 25% of the cycle while the falling edge comprised the remaining 75%. While we are certain that the proper data values were stored in the I/O buffer of the NI-DAQ card, we believe that the pattern generation scheme used by the card to output data may be creating data streams in a manner not suitable for our purposes. Further manipulations of the known data pattern stored in the buffer resulted in identical results.

The response of the BPSK to the voltage converter control as the TTL signal transitions can be seen in Figures 3 and 4. During testing, we found that noise from the power supplies was a major problem with this circuit, especially during the high-low transition. The effects of noise were limited by adding 100 nF coupling capacitors across each of the supplies.
Figure 3. Output of BPSK as input switches from low to high. Channel 1 shows the input to be modulated by the BPSK, Channel 2 shows the output of the BPSK transitioning from in-phase to out of phase, and Channel 3 shows the control signal from the voltage conversion circuit as it transitions from low to high.

Figure 4. Output of BPSK as input switches from high to low. Channel 1 shows the input to be modulated by the BPSK, Channel 2 shows the output of the BPSK transitioning from 180° out of phase to in-phase, Channel 3 shows the control signal from the voltage conversion circuit as it transitions from high to low.
6. Receiver

6.1 Design

The goal of the receiver was to convert the 5 ns, 200 mV avalanche pulse seen by Bob into a signal that could be read by the National Instruments I/O card. Our design used high-speed TTL components to perform this conversion; however, due to the short duration and low amplitude of the avalanche signal, there was no guarantee that conventional TTL circuits would operate fast enough to accomplish this goal. First, a comparator was used to detect the peak and produce TTL voltage levels. We then used a monostable multivibrator, or “one-shot,” circuit to lengthen the 5 ns pulse to 500 ns. The external timing components were chosen to produce a 500 ns output pulse with minimum retrigger time. To provide a theoretical retrigger time of approximately 90 ns, we chose an $R_{\text{EXT}}$ of 4.7 kΩ and a $C_{\text{EXT}}$ of 150 pF. The schematic is shown in Figure 5.

![Figure 5. Receiver hardware schematic.](image)

The design of the software interface was constrained by the same PCI bottlenecking issue faced by the transmitter, i.e. Bob could not be guaranteed uninterrupted control of the bus. To
solve this, a buffer was used to store the incoming bits. Two 74HC595 8-bit shift registers with output registers were used for the incoming bit buffer and the two chips were cascaded to allow for 16 bits of storage. The input of the buffer chip was serially shifted into the storage registers, then moved into the output registers. This allowed the storage registers to continue storing new data while the old data was read. Figure 6 shows a diagram of the buffer chip operation. A 74HC161 4-bit counter chip was used to count the number of clock cycles, or theoretically, photons, which arrived. This information is essential for proper reconciliation, as the exact number of bits received will vary between readings of the output registers. The complete buffer circuitry is shown in Figure 7.

Figure 6. 8-bit shift register with output registers.
6.2 Testing

After observing the response of the comparator to the avalanche input, we tested the retrigger time of the monostable multivibrator. To do this, we assembled the circuit on a protoboard and used a Hewlett Packard 33120A function generator to simulate a TTL input pulse. To test the minimum input pulse duration, we connected the comparator output to the multivibrator input. All outputs were monitored using a Tektronix JDS 3054 oscilloscope.

The serial input of the buffer was connected to a 0-5 V switchable source to simulate the output of the avalanche-detect circuitry. When the I/O card initiated a read operation, it pulsed the storage register clock, transferring the data to the output registers. The bits were then read by the I/O card while the flow of data to the storage registers remained uninterrupted.
6.3 Results

The high-speed comparator responded to a minimum avalanche duration of 5 ns. The output can be seen in Figure 8. With input from the function generator, the monostable multivibrator produced a longer output pulse than expected (600 ns), but was retriggerable at speeds up to 1.25 MHz. This result can be seen in Figure 9. When connected to the comparator, the one-shot circuit responded properly to a minimum avalanche duration of 10 ns, which was too long to be practical. The circuit response to an avalanche input can be seen in Figure 10.

Figure 8. Response of high-speed comparator (channel 3) to 5 ns avalanche signal (channel 1).
Figure 9. Response of monostable multivibrator (channel 2) to 1 MHz function generator input (channel 1).

Figure 10. Response of comparator (channel 2) and one-shot system (channel 3) to 10 ns avalanche input (channel 1).
With the two-chip buffer, the National Instruments I/O card could read 16 bits of data at 62.5 kHz (1/16 of 1 MHz), fast enough to meet our requirements. The actual frequency at which the bits were read was slightly faster to ensure that the buffer did not overflow before it was copied to the output registers; this frequency was still low enough that PCI bus availability was not a concern.

7. Conclusions

The team was able to implement a partial solution to the tasks proposed. Completion of the clock, base and bit generators was limited by the unexpected duty-cycle behavior of the I/O card. Conversion of the TTL signals was finalized. Little progress was made on the avalanche detection portion of the project due to specification delays and the difficulty encountered in finding components that met the timing requirements. The software interface was implemented and preliminary tests were performed; the buffer/registers are expected to meet the requirements of the system.

As anticipated, we encountered several technical challenges during the course of completing this project. On the software side, programming the National Instruments PCI-6534 I/O card and controlling it using C++ provided the largest challenge. Because the cards were designed to be used in a wide variety of applications, we had trouble adapting them to our specific purpose; we lost valuable time learning how to program the cards to meet our needs. Interfacing with the 68-pin PCI card connector provided difficulty during the software testing phase, as each pin had to be tediously counted and carefully connected. The original shift register design included only one 8-bit serial in/parallel out shift register. It was found that this was not sufficient to accurately transmit to the I/O card as the registers could continue to change
while being read, thus corrupting the data. To solve this problem, a second register was added in parallel. The delays in obtaining each of the chips, arising from the inadvertent ordering of surface mount rather than DIP packages, postponed the construction of the circuit and exhausted our time for software control and debugging.

On the transmitting hardware side, noise was an issue, though easily minimized by the use of coupling capacitors. On the receiving end, the largest problem was the speed of the avalanche pulse. Even ultra high-speed (5 ns) TTL components did not operate fast enough to respond to the 5 ns pulse. One solution to the avalanche detection problem is to use faster (2 ns) emitter-coupled logic (ECL) comparator and one-shot circuits to detect and hold the avalanche signal. An ECL to TTL converter would then be required for input to the shift registers. The GTL laboratory possesses an ECL monostable multivibrator, so it would not be difficult to perform further tests to determine the appropriateness of this solution. Further research needs to be conducted regarding the input voltage levels required for ECL components and the feasibility of ECL to TTL conversion.

Non-technical issues also played a role in delaying the completion of this project. The collaboration of the two very different Georgia Tech communities revealed the importance of good communication regarding project goals and expectations, especially on a shortened schedule. For example, important timing specifications should have been accurately defined initially, which would have avoided the ordering of incorrect components, and the resulting waste of time and money. This could have been easily avoided by including all members of the project in weekly meetings, rather than only the undergraduate students. Other factors in the delayed completion of this project included limited access to the GTL building and lab resources.
8. **Future Goals**

While our group made significant progress, work remains to be done to finalize this automated control system. Completion of this project hinges on conversion of the avalanche pulse to TTL voltage signals of sufficient duration. While it is hoped that ECL components will meet the requirements of the system, continued research and testing is required to develop an ideal solution. This project was one of many steps in the ongoing research of the GTL/CNRS Telecom laboratory. Continuing projects should investigate the efficiency of reconciliation algorithms as well as the possible detection and effects of an eavesdropper on the system.

9. **Economic Analysis**

Excluding the price of labor and of fabrication of the board, we estimate the total cost of each side (receiver and transmitter) to be less than $5, and is only this high because of the high-speed constraints on the components. We did not include the I/O card, BPSKs, and photo diode in our economic analysis since these components were provided by CNRS. Obviously, this system is not ready for mass production and thus it seems premature to put a total price on a finished QC system, but our contribution and suggestions for improving the current GTL-CNRS quantum cryptography scheme are very cost effective.

10. **References**


11. Acknowledgements

Thanks to Dr. Sayle, Dr. Smith, Dr. Rhodes, Dr. Malassanet, Olivier Guerreau, and the GTL/CNRS Telecom lab.