Quantum Circuits for Matrix Multiplication

Mario Sracic

Last updated July 26, 2011
## Contents

1 Definitions .............................................. 1

2 Motivation ............................................. 2

3 Quantum Circuits ...................................... 4
   3.1 General Multiplication for $\mathbb{Z}^+ \cup \{0\}$ .............. 5
   3.2 Row-Column Multiplier .................................. 7
   3.3 Row Adder ............................................. 7
   3.4 General Matrix Multiplication ......................... 8

4 Complexity Analysis .................................. 8

5 Conclusions ........................................... 10
List of Figures

1. NOT Gate ................................................................. 1
2. Three-bit Toffoli Gate, $T_3$ ........................................ 2
3. Controlled NOT Gate (XOR) .......................................... 3
4. Multiplication Circuit for $4 \cdot 3 \mod 6$ ............................ 6
5. General Multiplication Circuit ...................................... 11
6. Row Adder Circuit ...................................................... 12
7. Row/Column Multiplier Circuit .................................... 13
8. Circuit for General Matrix Multiplication .......................... 14
Abstract

In their paper *Quantum Networks for Elementary Arithmetic Operations*, Vedral, Barenco and Ekert introduced several circuits for basic arithmetic operations like addition modulo $N$, controlled multiplication and exponentiation. We would like to extend their multiplication circuit to perform general integer multiplication modulo $N$. Then, new designs for quantum circuits will be introduced that allow the construction of a quantum circuit that will implement general matrix multiplication. The goal is to provide the necessary stepping stones for the eventual breaking of the Sakalauskas, Tvarijonas, Raulynaitis Key Agreement Protocol (STR-KAP).
1 Definitions

Definition 1.0.1. Depth
The maximum length of any directed path from any input wire to any output wire

Definition 1.0.2. Entanglement
Quantum correlation between two observables [4]

Definition 1.0.3. \textit{NOT} Quantum Gate expressed as a unitary matrix

$$U_{\text{NOT}} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} |0\rangle$$

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{not_gate.png}
\caption{NOT Gate}
\end{figure}

Definition 1.0.4. Quantum Circuit
\begin{quote}
Let $H$ be a two-dimensional Hilbert Space. A Quantum circuit of width $n$ in $H$ is a unitary operator $U$ of size $2^n \times 2^n$ working on joint systems composed of $n$ qubits from $H$, which are initially in some base state of $H^n$. $U$ thus realizes some function $f : (\text{base of } H^n) \rightarrow H^n$. If we identify the base vectors of $H$, $|0\rangle$ and $|1\rangle$, with logical 0 and 1, and the output with some probability distribution by squaring the norms of amplitudes, we may say that for each bit string of length $n$, the circuit outputs some probability distribution over all bit strings. We may also treat some subsystems as auxiliary, in that case we forget about them when evaluating the output. [7]
\end{quote}

Definition 1.0.5. Quantum Gate
The analogue of a logic gate in a classical computer. [5]

Definition 1.0.6. qubit
A 2-dimensional, normalized, complex vector in a Hilbert space with base vectors $|0\rangle$ and $|1\rangle$. Each qubit can be represented as

$$|\Phi\rangle = a|0\rangle + b|1\rangle$$

with complex scalars $a$ and $b$ such that $\langle \Phi | \Phi \rangle = |a|^2 + |b|^2 = 1$.

Definition 1.0.7. Superposition
A qubit can be a linear combination of $|0\rangle$ and $|1\rangle$, or in general, a state of the form $c_0|0\rangle + c_1|1\rangle$ where $c_0, c_1$ are complex coefficients normalized to 1, i.e. $|c_0|^2 + |c_1|^2 = 1$.

Definition 1.0.8. Three-bit Toffoli Gate/Controlled-Controlled-NOT (CC-NOT or $T_3$)
\[ T_3 \]

<table>
<thead>
<tr>
<th>\langle 000 \rangle</th>
<th>\langle 001 \rangle</th>
<th>\langle 010 \rangle</th>
<th>\langle 011 \rangle</th>
<th>\langle 100 \rangle</th>
<th>\langle 101 \rangle</th>
<th>\langle 110 \rangle</th>
<th>\langle 111 \rangle</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 0 0 0</td>
<td>0 1 0 0</td>
<td>0 0 1 0</td>
<td>0 0 0 1</td>
<td>0 0 0 0</td>
<td>0 0 0 0</td>
<td>0 0 0 0</td>
<td>0 0 0 1</td>
</tr>
</tbody>
</table>

Figure 2: Three-bit Toffoli Gate, \( T_3 \)

Definition 1.0.9. **Unitarity**

The sum of all probabilities of all possible outcomes is one.

Definition 1.0.10. **Unitary Operator**

A linear transformation that is bijective and preserves length. \[4\]

Definition 1.0.11. **Walsh-Hadamaard Transform**:

\[
U_{W-H} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} |0\rangle \begin{bmatrix} |0\rangle \\ |1\rangle \end{bmatrix}
\]

Definition 1.0.12. **XOR Quantum Gate expressed as a unitary matrix**:

\[
U_{XOR} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} |00\rangle \\ |01\rangle \\ |10\rangle \\ |11\rangle \end{bmatrix}
\]

2 Motivation

In \[6\], Sakalauskas, Tvarijonas and Raulynaitis introduce a new key agreement protocol based on infinite non-abelian group presentation and representation levels. A short explanation of the protocol will be provided, however, it is left to the reader to refer to \[6\] for a more detailed explanation.
Consider two communicators, Alice and Bob who wish to exchange sensitive information over an unsecured line of communication. We will suppose that a third entity, named Eve, has managed to compromise the communications between Alice and Bob. In order to secure communications, Alice and Bob will create a key via the STR-KAP.

First, Alice and Bob agree on an infinite, non-abelian group \( \Gamma \) with a representation from \( \Gamma \to GL_d(F_{p^k}) \) denoted by \( \varphi \). Then they choose two mutually commuting subgroups \( A,B \in \Gamma \). Alice selects an \( \alpha \in A, \theta \in \Gamma \) and \( r \in \mathbb{N} \) and then forms \( \omega_1 = \alpha \theta \alpha^{-1} \) of which she maps to \( AQ \alpha A^{-1} \) via

\[
\varphi(\omega_1) = \varphi(\alpha \theta \alpha^{-1}) \\
= \varphi(\alpha) \varphi(\theta) \varphi(\alpha^{-1}) \quad \text{since } \varphi \text{ is a homomorphism} \\
= AQ \alpha A^{-1} \quad (1)
\]

Finally, Alice publishes \( AQ^r A^{-1} \). Meanwhile, Bob chooses an \( s \in \mathbb{N} \) and \( \beta \in B \), computing \( \omega_2 = \beta \theta \beta^{-1} \). After sending \( \omega_2 \) through \( \varphi \), he attains \( BQB^{-1} \) and finally publishes \( BQ^s B^{-1} \). Alice takes Bob’s result, raised to \( r \) and conjugates it by \( A \), resulting in

\[
A(BQ^s B^{-1})^r A^{-1} = A(BQ^s B^{-1}) A^{-1} = (AB)Q^{sr}(AB)^{-1} \quad (2)
\]

Bob raises Alice’s result by \( s \) and conjugates it by \( B \), reaching the same solution as in Eqn ( ), this is then multiplied by an \( a \in M \).

As mentioned earlier, the communication between Alice and Bob is not secure, there is the sinister third party Eve who wants to crack the key between Alice and Bob. However, the only information that Eve was able to intercept was \( Q, AQ^r A^{-1}, \) and \( BQ^s B^{-1} \). Therefore, in order to break the key, Eve must find some \( X,Y \in GL_d(F_{p^k}) \) and \( u,v \in \mathbb{N} \) such that

\[
(XY)Q^{uv}(XY)^{-1} = (AB)Q^{sr}(AB)^{-1}.
\]

Since Eve is required to multiply two matrices, it would be beneficial if she could check all possible products of all pairs of matrices. Therefore, harnessing the power of quantum computing to implement matrix multiplication would be imperative.
3 Quantum Circuits

The power of quantum computing is the ability to apply a quantum operation to a superposition of all possible inputs with an output of all possible outputs, i.e. *quantum parallelism*. There are specific properties that our *quantum gates* must satisfy, each gate must

- Be a unitary operator,
- reversible.

The simplest way to preserve unitarity in quantum gates is to build them from universal gates such as NOT, XOR and Toffoli or proven unitary transformations. Furthermore, we must abide by the following theorem

**Theorem 3.0.1. No Cloning Theorem**

*There cannot be a device that produces exact copies of a quantum state.*

Theorem 3.0.1 means that a circuit cannot take the state in a register and ‘copy’ that state to another register as you would copy a file. Instead, values can be replicated in another register by XORing the two registers.

A quantum circuit takes a “two state quantum system” called a qubit [2] which holds information in manner similar to the classical bits 0 and 1. The size of a circuit can be determined by the number of qubits necessary to perform the desired operation. For our purposes, we will define the number of qubits required by a number to be \( n = \lceil \log p \rceil \) when dealing with elements in \( \mathbb{F}_p \) and \( \eta = k \lceil \log p \rceil \) for elements in \( \mathbb{F}_{p^k} \). [3]

Vedral, Barenco and Ekert introduced quantum gates for elementary arithmetic operations like addition modulo \( N \) and a controlled multiplication circuit which is represented by the function \( f_{a,N}(x) = ax \mod N \). The controlled multiplication gates enacted the state change

\[
|c; x, 0 \rangle \rightarrow \begin{cases} 
|c; x, ax \mod N \rangle & \text{if } |c \rangle = |1 \rangle \\
|c; x, x \rangle & \text{if } |c \rangle = |0 \rangle 
\end{cases}
\]

on a quantum memory register; however, this circuit requires that the value to be multiplied, \( a \), be hardwired into the circuit, i.e. a new circuit would need to be constructed to multiply by different \( a \)'s. We seek an extension of the controlled multiplication circuit to allow the multiplication of any two integers modulo \( N \).
Consider $a, b \in \mathbb{Z}^+ \cup 0$ where $a, b$ are $n$-qubit numbers. Applying the binary expansion of $a$ and $b$ results in

$$a = a_n2^n + a_{n-1}2^{n-1} + a_{n-2}2^{n-2} + \cdots + a_12^1 + a_02^0$$

$$b = b_n2^n + b_{n-1}2^{n-1} + b_{n-2}2^{n-2} + \cdots + b_12^1 + b_02^0$$

(3)

Now, if we were to multiply $a$ and $b,$

$$a \cdot b = (a_n2^n + \cdots + a_12^1 + a_02^0) \cdot (b_n2^n + \cdots + b_12^1 + b_02^0)$$

$$= a_n2^n(b_n2^n + \cdots + b_12^1 + b_02^0) + a_{n-1}2^{n-1}(b_n2^n + \cdots + b_12^1 + b_02^0) + \cdots$$

$$+ a_12^1(b_n2^n + \cdots + b_12^1 + b_02^0) + a_02^0(b_n2^n + \cdots + b_12^1 + b_02^0)$$

$$= 2^n(a_nb_n) + \cdots + 2^{n-1}(a_nb_{n-1} + a_{n-1}b_1 + a_n b_0) + \cdots$$

$$+ 2^2(a_0b_2 + a_1b_1 + a_2b_0) + 2^1(a_0b_1 + a_1b_0) + 2^0(a_0b_0)$$

(4)

Notice that the indices of each pair of qubits $a_ib_j$ when added indicate the respective power of 2, i.e. $2^{i+j}$. This pattern is key in the design of a quantum circuit for general multiplication.

The general multiplication operation can be expressed as the state change

$$|c; a, b, 0 \rangle \rightarrow |c; a, b, a \cdot b \mod N \rangle,$$

where $a$ and $b$ are $n$-qubit numbers such that $a, b < N \leq 2^n, n = \lceil \log_2 N \rceil$ and $c$ is a control qubit. We maintain the inputs $a, b$ for the sake of the reversibility of the operation. Within the circuit, we have a temporary register of size $n$ to hold the powers of 2. It is important to notice that $2^k$ for $0 < k < 2n$ can exceed our modulus $N$. Therefore, we classically compute $2^k \mod N$ for all $k$ and store the values in the circuit. The general multiplication circuit can be seen in Figure 5 and proceeds as follows:

- Begin with $n$-qubit inputs $a$ and $b$ and examine each qubit individually. Utilizing the pattern discovered in Eqn [4], apply a three-qubit Toffoli Gate to the qubits $a_i, b_j$. If both qubits are 1 along with the control qubit, the temporary gate is activated and XORs the incoming value with $2^{i+j}$.
- The result is then fed into an addition modulo $N$ gate (designed by [2]) which adds the value to the third register which holds a running total.
- After the addition modulo $N$ gate, the temporary register is set to the state $|0\rangle$. 

5 of 14
Example 3.1.1. Let $a = 4, b = 3, N = 6$ where $a$ and $b$ are 3-qubit integers. Compute $2^k \mod 6$ for $0 \leq k \leq n$ and input the results into the circuit as seen in Figure 4.

$$
2^0 \equiv 1 \mod 6 \\
2^1 \equiv 2 \mod 6 \\
2^2 \equiv 4 \mod 6 \\
2^3 = 8 \equiv 2 \mod 6
$$

Then the general multiplication algorithm is as follows:

i Input $a = 4$ and $b = 3$ into registers one and two respectively and load $N = 6$ into the addition modulo $N$ circuit and set the control qubit to 1.

ii $|1; 4, 3, 0, 0\rangle \rightarrow |1; 4, 3, 0, 0\rangle$

iii $|1; 4, 3, 0, 0\rangle \rightarrow |1; 4, 3, 0, 0\rangle \rightarrow |1; 4, 3, 0, 0\rangle$

iv $|1; 4, 3, 0, 0\rangle \rightarrow |1; 4, 3, 0, 0\rangle \rightarrow |1; 4, 3, 4, 4\rangle \rightarrow |1; 4, 3, 0, 4\rangle$

v $|1; 4, 3, 0, 4\rangle \rightarrow |1; 4, 3, 0, 4\rangle \rightarrow |1; 4, 3, 2, 0\rangle \rightarrow |1; 4, 3, 0, 0\rangle$

vi The resulting state is $|1; 4, 3, 0, 0\rangle$, so the answer to $4 \cdot 3 \mod 6$ is 0.

Figure 4: Multiplication Circuit for $4 \cdot 3 \mod 6$
3.2 Row-Column Multiplier

With a general multiplication circuit possible, we are able to begin constructing more complex circuits to implement different aspects of matrix multiplication. A new circuit, called the **Row-Column Multiplier**, performs the multiplication of rows and columns.

Consider two $d \times d$ matrices $A, B \in \text{GL}_d(\mathbb{F}_{p^k})$. The row-column multiplication circuit multiplies a row $R_i \in A$ with a column $C_j \in B$, performing the state change

$$|R_i, C_j, 0\rangle \rightarrow |R_i, C_j, R_iC_j\rangle,$$

where $R_i$ and $C_j$ are $d\eta$ qubits each and the third register is of the same size. Recall that $A, B \in \text{GL}_d(\mathbb{F}_{p^k})$, so their entries come from $\mathbb{F}_{p^k}$ and so the number of qubits used by an entry is $\eta$. Figure 7 illustrates the circuit,

- First, the rows of matrix $A$ are the inputs for the first register and the columns of matrix $B$ are inputs of the second register. The appropriate values, $a_{ij}$ and $b_{ji}$ are then sent as inputs to the general multiplication gate. Meanwhile, the third register, initialized to all zeros, sends an entry to hold the product of $a_{ij}b_{ji}$.

- The general multiplication gate returns in the inputs $a_{ij}$ and $b_{ji}$ for $1 \leq i, j \leq d$, therefore maintaining the integrity of the original inputs.

- At the end of the circuit, $R_i$ and $C_j$ are reassembled and the third register now holds the products $R_iC_j$, for $1 \leq i, j \leq d$.

3.3 Row Adder

The second half of matrix multiplication is the summation of the products from a row times a column. Therefore, we wish to perform the state change

$$|c; R_iC_j, 0\rangle \rightarrow |c; R_iC_j, c_{ij}\rangle.$$

We have $R_iC_j$ as a $d\eta$ qubit register and the second as a $\eta$-qubit register. Figure 6 is the **Row Adder** circuit and the circuit proceeds as follows:

- Once some $R_i \in A$ and $C_j \in B$ has been inputed into the Row-Column Multiplier, we take the result and use it as the input for the Row Adder. Applying an addition modulo $N$ gate to each subsequent entry for a total of $d - 1$ gates, results in the entry $c_{ij}$.

- A controlled $n$-qubit Toffoli gate to applied to the last entry in order XOR the result with the second register.

- Then apply a reverse modulo $N$ adder working upwards to undo the addition modulo $N$ in order to achieve the original state of the input, $R_iC_j$. 
3.4 General Matrix Multiplication

With our Row-Column Multiplier and Row Adder Circuits, we can utilize them to construct a general purpose matrix multiplication circuit. Figure 8 is the diagram for the matrix multiplication circuit which implements the following state change on the memory register

\[ |A, B, 0, 0\rangle \rightarrow |A, B, R_iC_j, A \cdot B \mod N\rangle. \]

We take the resulting matrix modulo \( N \) to prevent memory overflow within the register.

The first two input registers are the matrices to be multiplied which are of size \( d^2 \eta \) qubits, the third register holds the products from the row-column multiplication and consists of a matrix with product strings in each entry. Thus the third register is \( d^3 \eta \) qubits in size. The fourth register will hold the product of the two matrices and so is \( d^3 \eta \) in size. Following Figure 8, the circuit proceeds in the following manner:

- Matrices \( A \) and \( B \) are set as inputs, then separate \( A \) into each row, \( R_i \) for \( 1 \leq i \leq d \), and \( B \) into columns, \( C_j \) for \( 0 \leq j \leq d \). Perform a Row-Column Multiply with each combination of \( R_i \) and \( C_j \) with the result being stored in the entries of the third register.

- Next, send each entry in the third register through a Row Adder Gate, storing the result in the fourth register modulo \( N \).

- Finally, we reconstruct the matrices \( A \) and \( B \) along with \( R_iC_j \) stored in the third register. The fourth register holds the new matrix which is the product of \( A \cdot B \mod N \).

Since the Row-Column Circuit and Row Adder are unitary operators, the general matrix multiplier is also unitary. Further, by the use of four registers, we are able to reverse the circuit.

4 Complexity Analysis

We will analyze each circuit based upon three criteria,

- The total number of qubits used to stored the relevant information.
- The number of elementary gates used within the circuit. The abbreviations are
  - \( N \): NOT Gate
  - \( CN \): Controlled NOT
  - \( C^2N \): Controlled-Controlled NOT
  - \( C^3N \): Controlled-Controlled-Controlled NOT
- The depth of the circuit (see Defn (1.0.1)).
We provide the complexity of the addition modulo $N$ circuit found in [2] since our circuits depend on this particular circuit. Further, since the primary goal of our circuits corresponds to the STR-KAP, we will assume our elements come from $\mathbb{F}_{p^k}$.

- **Addition Modulo $p$**
  - Number of qubits: $3\eta + 2$
  - Number of Gates: $(20\eta + 10)C^2N + (20\eta + 24)CN + 2N$
  - Depth: 29

- **General Purpose Multiplier for $\mathbb{Z}^+ \cup 0$**
  Since the number of terms in each power of 2 is $i + j + 1$ for
  - Number of qubits: $5\eta + 2$
  - Number of Gates: $(2\eta^2 + 4\eta + 2)C^3N + (20\eta^3 + 50\eta^2 + 40\eta + 10)C^2N + (20\eta^3 + 64\eta^2 + 68\eta + 24)CN + (2\eta^2 + 4\eta + 2)N$
  - Depth: $29\eta^2 + 58\eta + 29$

- **Row-Column Multiplier**
  - Number of qubits: $3\eta$
  - Number of Gates: $(d - 1)(2\eta^2 + 4\eta + 2)C^3N + (d - 1)(20\eta^3 + 50\eta^2 + 40\eta + 10)C^2N + (d - 1)(20\eta^3 + 64\eta^2 + 68\eta + 24)CN + (d - 1)(2\eta^2 + 4\eta + 2)N$
  - Depth: $29\eta^2 + 58\eta + 29$

- **Row Adder**
  - Number of qubits: $d\eta + 2\eta + 1$
  - Number of Gates: $[(2d - 2)(20\eta + 10) + \eta]C^2N + (2d - 2)(20\eta + 24)CN + (4d - 4)N$
  - Depth: $116 + \eta$

- **General Purpose Multiplier for Matrices**
  - Number of qubits: $d^3\eta + 3d^2\eta$
  - Number of Gates:
    
    $$(d^3 - d^2)(2\eta^2 + 4\eta + 2)C^3N + (d^3 - d^2)(20\eta^3 + 50\eta^2 + 40\eta + 10) + [(2d^3 - 2d^2)(20\eta + 10) + d^2\eta])C^2N + (d^3 - d^2)(20\eta^3 + 64\eta^2 + 68\eta + 24) + (2d^3 - 2d^2)(20\eta + 24))CN + (d^3 - d^2)(2\eta^2 + 4\eta + 2) + (4d^3 - 4d^2))N$$
  - Depth: $29d\eta^2 + 58d\eta + 29d$
5 Conclusions

We have taken the first steps in the long process of finding a method to break the STR-KAP. The General Matric Multiplication circuit could me used in a brute force attack on the Key Agreement Protocol; however, in cite george’s paper, it was proven that a brute force attack is not that efficient at finding the order of a matrix. In the future, more research will be done to find a method to speed up the matrix multiplication circuit, potentially through the use of the Quantum Fourier Transform. Furthermore, we will look into the construction of quantum circuits to implement other linear algebra operations like Cramer’s Rule and row operations, including a circuit for determinants.

References


Figure 5: General Multiplication Circuit
Figure 6: Row Adder Circuit
Figure 7: Row/Column Multiplier Circuit
Figure 8: Circuit for General Matrix Multiplication