Ataques quânticos ao gerador pseudo-aleatório de Blum-Micali

Nigini A. Oliveira, Edmar J. Nascimento e F. M. Assis.

***Resumo*— A utilização de seqüências pseudo-aleatórias é de grande importância em diversas aplicações da criptografia. Um ataque a um gerador pseudo-aleatório objetiva reproduzir uma dada seqüência pseudo-aleatória a partir de um conjunto de termos dessa seqüência. Classicamente, esse é um problema de difícil solução, ou seja, não existe algoritmos conhecidos em tempo polinomial que realizem esta tarefa. Por outro lado, o desenvolvimento de algoritmos quânticos vem mostrando que determinados tipos de problemas podem ser resolvidos de modo mais eficiente. Neste artigo, é apresentado um circuito quântico que realiza um ataque ao gerador Blum-Micali apresentando um ganho no número de operações em relação aos algoritmos clássicos conhecidos.**

***Palavras-Chave*— Gerador Blum-Micali, algoritmos quânticos, seqüências pseudo-aleatórias.**

***Abstract*— Pseudo-random sequences are very important to a wide range of applications in cryptography. The purpose of attacking a pseudo-random sequence is to predict it from a smaller set of terms. Classically, this is a problem that is hard to solve. This means that no polynomial-time algorithms are available. On the other hand, quantum algorithms are proved to solve some computation problems in a more efficient way. In this article, we present a quantum circuit that attacks Blum-Micali's generator providing computational gains over classical counterpart solutions.**

***Keywords*— Blum-Micali's generator, quantum algorithms, pseudo-random sequences.**

I. INTRODUÇÃO

A utilização de seqüências de números aleatórios se faz necessária em diversas aplicações da criptografia tais como: esquemas de autenticação recíprocos, geração de chaves de sessão e a geração de chaves para o algoritmo de chave pública RSA [1]. As seqüências de números aleatórios devem atender a dois requisitos principais: aleatoriedade e imprevisibilidade. Para que a aleatoriedade seja garantida, os símbolos da seqüência devem ser uniformemente distribuídos e independentes. A imprevisibilidade garante que um termo da seqüência não pode ser inferido a partir de termos anteriores. Para aplicações práticas, é bastante difícil conseguir fontes de números verdadeiramente aleatórios. Dessa forma, recorre-se a algoritmos determinísticos para a geração de seqüências que satisfazem alguns testes estatísticos. Essas seqüências são chamadas de pseudo-aleatórias.

Os geradores de seqüências pseudo-aleatórias atuam de modo bastante similar. Em geral, é realizado algum tipo de operação iterativa em aritmética modular, em que é usado como módulo um número primo muito grande. Dentre esses geradores, o gerador de Blum-Micali é um dos mais simples. A sua segurança é baseada na intratabilidade computacional do problema do logaritmo discreto [5]. Classicamente, a dificuldade desse problema parece ser da mesma ordem de magnitude que a fatoração de números primos requerida para o sistema RSA [1].

Por outro lado, existem algoritmos quânticos que se destacam na solução de problemas computacionais difíceis, que não possuem solução em tempo polinomial. Particularmente, os algoritmos de Grover, Shor, Deutsch e a transformada de Fourier quântica proporcionam ganhos consideráveis ao se comparar com os algoritmos clássicos correspondentes [2], [3]. Com a aplicação da transformada de Fourier quântica, consegue-se resolver o problema do logaritmo discreto com $\mathcal{O}(\lceil \log r \rceil)^2$ operações, em que r é o menor número inteiro tal que $a^r \mod p = 1$. Uma das razões para o ganho computacional proporcionado pelos algoritmos quânticos é devido a uma propriedade conhecida como *superposição quântica*. A superposição quântica permite codificar todos os valores do domínio de uma dada função em um único estado quântico composto por *qubits* (bits quânticos). Ao realizar uma transformação sobre o estado quântico em superposição, realiza-se a avaliação simultânea de todos os valores do domínio da função a partir de uma única operação.

Motivado pelas boas perspectivas oriundas dos algoritmos quânticos, é proposto neste artigo um circuito quântico que permite encontrar a chave secreta do gerador Blum-Micali a partir da seqüência pseudo-aleatória gerada e dos parâmetros públicos do gerador. Esse circuito proporciona um ganho no número de operações realizadas. Este artigo está organizado da seguinte maneira. Na seção II, é descrito o funcionamento do gerador Blum-Micali e também são apresentados alguns resultados referentes à sua segurança. Na seção III, é apresentada uma revisão sobre alguns tópicos da mecânica quântica necessários ao entendimento da solução apresentada nesse artigo. Na seção IV, é apresentado o circuito quântico proposto nesse artigo para o ataque ao gerador Blum-Micali.

II. GERADOR DE BLUM-MICALI

O gerador Blum-Micali (BM) é definido da seguinte forma [7]. Seja p um número primo, escolhe-se g como sendo um gerador para $\mathbb{Z}_p^*$ e a semente, a chave secreta do gerador, x_0. Os valores de p e g podem ser publicamente conhecidos e usados diversas vezes seguidas. A geração dos bits da seqüência pseudo-aleatória é feita da seguinte forma: para o

Nigini A. Oliveira, Departamento de Sistemas e Computação. Edmar J. Nascimento e F. M. Assis, Departamento de Engenharia Elétrica. Universidade Federal de Campina Grande, Campina Grande-PB, Brasil, E-mails: nigini@dsc.ufcg.edu.br, jnedmar@dee.ufcg.edu.br, fmarcos@dee.ufcg.edu.br.
Este trabalho foi parcialmente financiado pelo CNPq

i-ésimo bit b_i, começando com $i = 1$, tem-se:

$$x_i = g^{x_{i-1}} \mod p; \quad (1)$$

$$b_i = \delta_{x_i > (p-1)/2}. \quad (2)$$

Em que a equação (2) indica que $b_i = 1$ se e somente se $x_i > (p-1)/2$.

A prova de que o gerador BM é computacionalmente seguro é baseada na hipótese da intratabilidade computacional do problema do logaritmo discreto. Blum e Micali mostraram que admitindo-se a hipótese precedente, não é possível calcular em tempo polinomial o bit b_i a partir do valor de x_{i+1}. Esse problema é conhecido como *hard-core bit problem.*

III. Tópicos gerais da mecânica quântica

Os sistemas quânticos são descritos pelas leis da mecânica quântica e esta, por sua vez, se baseia num conjunto de postulados que servem de base para toda a teoria quântica [4]. Um sistema quântico isolado é descrito por um vetor de estados unitário $|\psi\rangle$ em um espaço de Hilbert $\mathcal{H}$. O sistema quântico mais simples é o *qubit* (bit quântico), um sistema quântico associado a um espaço de Hilbert de dimensão dois. De acordo com a notação de Dirac, um qubit pode ser representado como

$$|\psi\rangle = a|0\rangle + b|1\rangle. \quad (3)$$

Sendo que $|0\rangle = [1\ 0]^T$ e $|1\rangle = [0\ 1]^T$ formam uma base ortonormal para o espaço de estados do sistema conhecida como a base computacional. a e b são números complexos arbitrários que satisfazem a relação de normalização $|a|^2 + |b|^2 = 1$. A equação (3) pode ser interpretada como uma superposição dos estados $|0\rangle$ e $|1\rangle$, em que esses estados tem respectivamente as probabilidades $|a|^2$ e $|b|^2$ de serem observados após uma medida de $|\psi\rangle$ na base computacional.

A evolução de sistemas quânticos isolados é descrita por meio de transformações unitárias, de forma que o vetor de estados $|\psi\rangle$ no instante de tempo t_1 está relacionado com o vetor de estados $|\psi'\rangle$ no instante de tempo t_2 pela relação

$$|\psi'\rangle = U|\psi\rangle. \quad (4)$$

Sendo que U é um operador unitário que depende apenas dos instantes t_1 e t_2.

As medidas quânticas são descritas através de *observáveis*. Um observável é uma propriedade de um sistema físico que, em princípio, pode ser medida como por exemplo, a posição de uma partícula. Na mecânica quântica, um observável é representado por um operador auto-adjunto $\mathbf{A}$, em que $\mathbf{A} = \mathbf{A}^\dagger$. Um operador auto-adjunto em um espaço de Hilbert $\mathcal{H}$ possui uma representação espectral, ou seja, os seus auto-estados formam uma base ortonormal completa em $\mathcal{H}$. Um operador auto-adjunto $\mathbf{A}$ pode ser então expressado como

$$\mathbf{A} = \sum_n \lambda_n \mathbf{P}_n. \quad (5)$$

Em que λ_n é um autovalor de $\mathbf{A}$ e $\mathbf{P}_n$ é um projetor ortogonal no espaço dos autovetores com autovalor λ_n. Ao medir um sistema quântico, a saída numérica do processo de medição de um observável $\mathbf{A}$ é um autovalor λ_n de $\mathbf{A}$. O estado do sistema é modificado, de modo que logo após a medição, o estado quântico passa a ser um auto-estado do observável $\mathbf{A}$ correspondente ao autovalor medido. Se o estado quântico imediatamente antes da medida é $|\psi\rangle$, então a saída λ_n é obtida com probabilidade

$$P(\lambda_n) = \| \mathbf{P}_n|\psi\rangle \|^2 = \langle\psi|\mathbf{P}_n|\psi\rangle. \quad (6)$$

Se a saída λ_n é obtida, o estado normalizado pós-medição é

$$\frac{\mathbf{P}_n|\psi\rangle}{\sqrt{\langle\psi|\mathbf{P}_n|\psi\rangle}}. \quad (7)$$

A medida do estado $|\psi\rangle$ dado pela equação (3) com o observável $A = |0\rangle\langle 0| - |1\rangle\langle 1|$ resulta em $+1$ com probabilidade $|a|^2$ e -1 com probabilidade $|b|^2$. Se o resultado da medida for $+1$, o estado pós-medição é $|0\rangle$. Se o resultado da medida for -1, o estado pós-medição é $|1\rangle$.

A. Circuitos Quânticos

Os algoritmos quânticos são implementados através de circuitos quânticos. Nos circuitos quânticos, os qubits são os elementos de informação sobre os quais são realizadas operações a fim de obter um resultado desejado. As transformações realizadas sobre os qubits são descritas por meio de transformações unitárias implementadas pelas portas quânticas. Uma porta quântica de grande importância é a porta de Hadamard. A porta de Hadamard é descrita pela operação unitária

$$H \equiv \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}. \quad (8)$$

Ao atuar nos estados da base computacional $|0\rangle$ e $|1\rangle$, a porta de Hadamard realiza as seguintes transformações

$$|0\rangle \xrightarrow{H} \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \equiv |+\rangle, \quad (9)$$

$$|1\rangle \xrightarrow{H} \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \equiv |-\rangle. \quad (10)$$

Ou seja, a sua aplicação cria uma superposição de estados a partir dos estados base $|0\rangle$ e $|1\rangle$. A aplicação de portas de Hadamard nos qubits individuais de um sistema composto de N qubits gera uma superposição de 2^N termos. Por exemplo, para o estado $|\psi\rangle = |0\rangle^{\otimes N}$, a aplicação de $H^{\otimes N}$ resulta em

$$H^{\otimes N}|0\rangle^{\otimes N} = \frac{1}{2^{n/2}} \sum_{x_i \in \{0,1\}^n} |x_i\rangle. \quad (11)$$

A superposição gerada pela equação (11) permite gerar a entrada para circuitos quânticos mais complicados como o que é proposto neste artigo.

B. Algoritmo de Grover

O algoritmo de Grover [8], ou algoritmo quântico de busca, utiliza-se da superposição descrita pela equação (11) para realizar a busca de M elementos em um conjunto desordenado de N elementos. Para isso, é necessário a construção de uma transformação unitária denominada oráculo (O). Esta porta é responsável por marcar os elementos buscados dentro do conjunto completo de elementos, que por sua vez estão codificados em uma superposição.

Um exemplo dessa operação é a busca do elemento 1 no conjunto $\{0, 1, 2, 3\}$, o qual pode ser codificado através de uma superposição de dois qubits. Para este exemplo, o estado inicial do sistema pode ser representado pelo estado $|\Psi\rangle = |001\rangle$, em que os dois primeiros qubits são usados para codificar os quatro valores possíveis do espaço de busca e o terceiro é utilizado pelo oráculo de Grover para a marcação do elemento buscado. A aplicação do operador Hadamard (8) em cada um dos três qubits de $|\Psi\rangle$ cria um estado $|\Psi_1\rangle = 1/2(|00\rangle + |01\rangle + |10\rangle + |11\rangle) \otimes |-\rangle$. É sobre este estado que o oráculo deve ser aplicado. O oráculo necessário para efetuar a marcação do valor 1 no estado $|\Psi_1\rangle$ é obtido através de uma porta quântica CNOT (NOT controlado) no circuito quântico indicado na Figura 1. A aplicação desse circuito inverte a fase do terceiro qubit, transforma $|-\rangle$ em $|+\rangle$, apenas para o termo $|01\rangle$ da superposição.

Fig. 1. Circuito quântico para a marcação do elemento 1 codificado pelo elemento $|01\rangle$ da superposição.

O mais importante a se entender é que o resultado desta operação inverte, a grosso modo, o sinal do elemento procurado. Um engano comum é pensar que a leitura dos qubits neste ponto resulta no elemento marcado. Na verdade, o elemento marcado teve apenas uma inversão de fase e não de sua amplitude probabilística com relação ao restante dos elementos do conjunto. Por isso, uma terceira e última fase é necessária ao algoritmo de Grover, a amplificação de fase realizada pela transformação $2|\Psi\rangle\langle\Psi| . A amplificação de fase é responsável por aumentar a probabilidade de leitura daqueles valores que receberam a marcação.

Para uma maior probabilidade de leitura dos elementos marcados, a aplicação do oráculo seguida pela amplificação deve ser executada algumas vezes. Essa seqüência iterativa é chamada de iteração de Grover ($G = 2|\Psi\rangle\langle\Psi| O$) e a quantidade de vezes que ela deve ser executada para levar a maior probabilidade possível depende apenas de M e N. Vários estudos já foram feitos a respeito e prova-se que a quantidade de iterações é da ordem de $\mathcal{O}(\sqrt{N/M})$[9].

IV. Busca quântica de X_i

Nesta seção são analisadas algumas características de um ataque quântico ao gerador BM descrito na seção II. A idéia base do algoritmo é a reconstrução da palavra gerada b, testando as possíveis seqüências geradoras, partindo-se do conjunto com todos os possíveis x_0 e reduzindo-se o mesmo levando em consideração os bits b_i. Obviamente que no caso clássico esta solução força-bruta não leva a bons resultados devido à explosão exponencial de possibilidades. No caso da utilização de um computador quântico, pode-se lançar mão do seu paralelismo para sanar tal dificuldade.

Nas análises de segurança, pressupõe-se que o atacante do gerador BM tem conhecimento dos seguintes parâmetros:

- O módulo primo p;
- O gerador do grupo finito g;
- Uma palavra pseudo-aleatória $b = b_1 b_2 \cdots b_l$ gerada pelo gerador BM.

O circuito de ataque proposto é mostrado na Figura 2. Esse circuito possui como entrada quatro registradores e os seus blocos estão divididos em quatro fases de processamento. O primeiro registrador possui os bits b_i da palavra gerada, codificados através dos estados quânticos $|b_i\rangle$, enquanto o segundo possui $\lceil \log p \rceil$ qubits, necessários para gerar os elementos x_i calculados pela equação (1). O terceiro e o quarto registradores são compostos por qubits auxiliares, sendo que o quarto possui apenas um qubit utilizado pelo algoritmo de Grover e o terceiro recebe tantos qubits quantos b_is forem utilizados no primeiro registrador.

As quatro etapas do circuito realizam as seguintes funções:

1) Preparação da superposição a partir dos qubits do segundo registrador com portas Hadamard;
2) Iterações de filtragem (portas $F(b_i)$) e permutação (portas P) dos elementos do segundo registrador, marcando o resultado no terceiro registrador;
3) Aplicação de $\sqrt{p}$ iterações de Grover (porta G^*), para amplificar o elemento marcado;
4) Leitura dos qubits do segundo registrador.

A seguir, a segunda e a quarta etapa do circuito são detalhadas.

A. Segunda Etapa - Filtragem e Permutação

Esta segunda etapa do algoritmo simula o ataque clássico força-bruta no que diz respeito a aplicar o algoritmo de Blum-Micali em todos os elementos do grupo. A diferença clara é que a utilização da superposição quântica torna estes cálculos muito mais eficientes. Assim como o gerador pseudo-aleatório, o circuito mostrado é iterativo, buscando filtrar, dentro do conjunto de possibilidades, aqueles elementos capazes de gerar o bit b_i em questão.

Esta etapa do circuito é montada com dois tipos de porta. $F(b_i)$ é um filtro dos elementos que podem ter gerado o bit b_i segundo o algoritmo de Blum-Micali. Assim, a função desta porta é apenas testar se os elementos da superposição são maiores ou menores que $l = (p-1)/2$. Caso b_i seja 0 e o elemento superposto menor que l, então um bit específico no terceiro registrador será invertido. O mesmo bit será invertido no caso em que b_i seja 1 e o elemento maior que l.

A segunda porta usada nesta fase é denominada P. Esta opera apenas sobre o segundo registrador, porém, controlada pelos elementos do terceiro. A operação realizada é a permutação dos elementos marcados simulando a geração de cada bit da palavra b_i. Uma permutação nos moldes do algoritmo de BM pode ser implementado por um somatório de projetores.

B. Quarta Etapa - Amplificação e leitura

Esta etapa do algoritmo é composta pela aplicação do algoritmo de Grover já explicada na seção III-B. Para isso, o oráculo a ser utilizado no circuito proposto como solução é definido pelo circuito indicado a seguir:

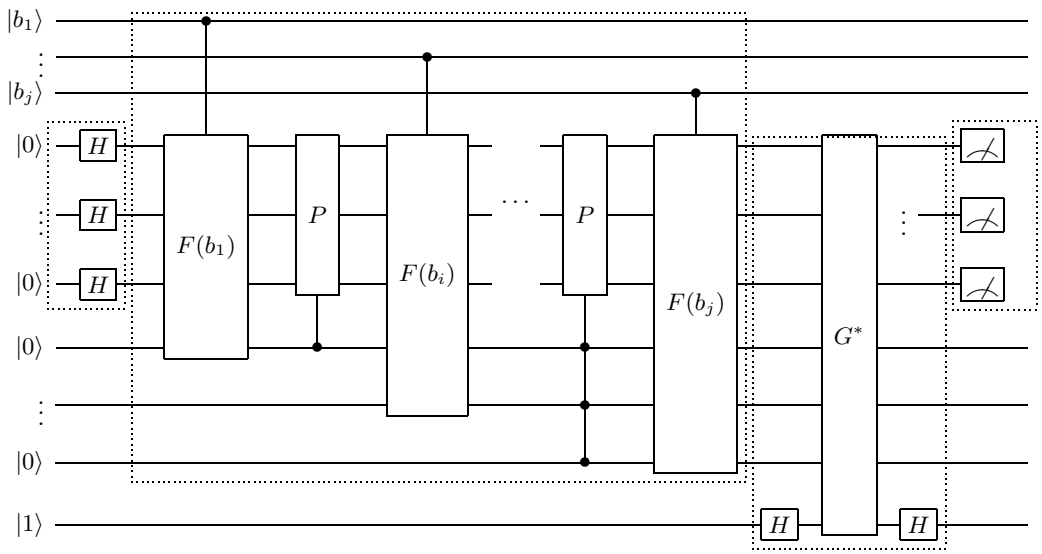
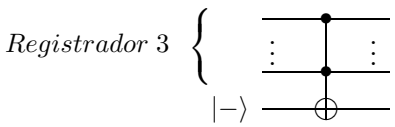


Fig. 2. Circuito de ataque ao gerador BM. As quatro fases do circuito estão destacadas nos retângulos tracejados da direita para a esquerda.



Significando que o elemento do segundo registrador, associado ao valor $|11\ldots1\rangle$ no terceiro registrador terá sua probabilidade de leitura amplificada a um valor muito próximo de 1.

Como já definido anteriormente, o algoritmo quântico de busca executa $\mathcal{O}(\sqrt{N/M})$ chamadas ao oráculo, sendo que no caso do circuito proposto, $M = 1$ e $N = \log p$. Desta forma, o circuito será capaz de ler um valor x_i, com alta probabilidade, após $\mathcal{O}(\sqrt{\log p})$ chamadas ao oráculo.

Com a aplicação desse algoritmo, o gerador BM tem um estado interno x_i identificado a partir de valores disponíveis a qualquer atacante. Com este valor, é possível simular deterministicamente o funcionamento do gerador, quebrando a propriedade de pseudo-aleatoriedade do mesmo. Um último passo pode ainda ser realizado para a descoberta do valor da chave secreta x_0. Para isso é utilizado o cálculo do logaritmo discreto, que é um problema tratável através da transformada quântica de Fourier (ver seção I).

C. Exemplos

1) Exemplo 1: Considera-se um gerador BM definido por $p = 7$ e $g = 3$. A seqüência binária gerada pelo gerador é $b = 10$. Desta forma o circuito apresentado necessita de dois qubits no primeiro registrador, $\lceil \log 7 \rceil = 3$ no segundo para representar os números de um a sete em binário, e mais três qubits auxiliares. A entrada do circuito pode ser escrita no estado $|\Psi\rangle = |10\rangle \otimes |000\rangle \otimes |00\rangle \otimes |1\rangle$. Após a aplicação das portas Hadamard na primeira fase do circuito, o estado inicial do algoritmo sem normalização é descrito por:

$$\begin{aligned} |\Psi_0\rangle &= |10\rangle \otimes (|000\rangle + |001\rangle + |010\rangle + |011\rangle + \\ &\quad |100\rangle + |101\rangle + |110\rangle + |111\rangle) \otimes \\ &\quad |00\rangle \otimes |1\rangle \end{aligned} \tag{12}$$

.

A aplicação da operação de filtragem a partir do primeiro qubit $|b_1\rangle = |1\rangle$ no estado $|\Psi_0\rangle$ resulta em:

$$\begin{aligned} |\Psi_1\rangle &= F(1)\,|\Psi_0\rangle = |10\rangle \otimes \\ &\quad [(|000\rangle + |001\rangle + |010\rangle + |011\rangle)\,|00\rangle) + \\ &\quad (|100\rangle + |101\rangle + |110\rangle + |111\rangle)\,|10\rangle] \otimes |1\rangle \end{aligned} \tag{13}$$

. Observa-se então que os elementos do segundo registrador que são maiores que $l = (7-1)/2 = 3$ foram associados ao estado $|10\rangle$ no terceiro registrador. Leva-se em consideração aqui, o fato de que a implementação de um circuito clássico para o teste *maior-que-menor-que* é trivial e eficiente. Assim, a implementação da porta F através de um circuito quântico é provadamente possível e eficiente, como mostrado em [10].

A operação de permutação é obtida a partir da tabela de permutação do grupo $\mathbb{Z}_p^*$. Para os estados que não estão nesse grupo, a permutação leva esses estados para eles mesmos. Sendo assim, P é dado por (qubits representados em notação decimal):

$$\begin{aligned} P &= |0\rangle\langle 0| + |5\rangle\langle 1| + |2\rangle\langle 2| + |6\rangle\langle 3| + \\ &\quad |4\rangle\langle 4| + |5\rangle\langle 5| + |1\rangle\langle 6| + |7\rangle\langle 7|\,. \end{aligned} \tag{14}$$

A aplicação de P em $|\Psi_1\rangle$ leva o sistema ao estado $|\Psi_2\rangle$ dado por:

$$\begin{aligned} |\Psi_2\rangle &= P\,|\Psi_1\rangle = |10\rangle \otimes \\ &\quad [(|000\rangle + |001\rangle + |010\rangle + |011\rangle)\,|00\rangle) + \\ &\quad (|100\rangle + |101\rangle + |001\rangle + |111\rangle)\,|10\rangle] \otimes |1\rangle\,. \end{aligned} \tag{15}$$

Neste caso, apenas o elemento $|110\rangle$ é permutado para o valor $|001\rangle$, pois o operador P é controlado pelo primeiro qubit do terceiro registrador e as outras componentes ($|100\rangle$ e $|101\rangle$) são permutadas para o mesmo valor. Aplicando novamente a operação de filtragem, agora levando em conta o qubit $|b_2\rangle =$

$|0\rangle$, tem-se o novo estado do sistema:

$$\begin{aligned} |\Psi_3\rangle &= F(0)\,|\Psi_2\rangle = |10\rangle \otimes \\ &[(|000\rangle + |001\rangle + |010\rangle + |011\rangle)\,|00\rangle) + \\ &(|100\rangle + |101\rangle + |111\rangle)\,|10\rangle + \\ &(|001\rangle)\,|11\rangle] \otimes |1\rangle\,. \end{aligned} \tag{16}$$

Percebe-se que um único elemento está associado ao valor $|11\rangle$ no terceiro registrador, significando que a etapa de filtragem foi concluída.

2) Exemplo 2: Considera-se agora que o gerador BM é definido por $p = 19$ e $g = 2$ e a seqüência binária é dada por $b = b_1b_2b_3 = 101$. Como a quantidade de qubits no segundo registrador é $k = \lceil \log p \rceil = 5$, a superposição gerada pelas portas Hadamard compreende a faixa de valores $\{0, 1, \cdots, 31\}$. Porém, os números válidos para esse gerador BM pertencem à faixa $\{1, 2, \cdots, 18\}$. Esses valores presentes no estado $|\Psi_0\rangle$ são eliminados pelas próprias iterações de execução do circuito apresentado, ou ainda, pode-se considerar um pequeno passo de marcação (similar à porta F) que elimina todos os valores maiores que $p - 1$. Além disso, deve-se também considerar a necessidade de construir a porta P de forma que possua a projeção de todos os valores do conjunto: $|19\rangle\langle 19| + \ldots + |32\rangle\langle 32|$, para que o operador seja unitário.

Na execução do algoritmo, o teste relativo ao bit b_1 marcará o grupo de elementos $\{10, \ldots, 18, 19, \ldots, 32\}$, pois apenas estes elementos são maiores que $l = 9$. A porta P permuta estes elementos e gera um novo sub-conjunto igual a $\{5, 3, 6, 1, 10, 12, 15, 11, 17, 19, \ldots, 32\}$. O teste para o segundo bit marcará apenas os valores $\{5, 3, 6, 1\}$, eliminando os números indesejados do conjunto inicial. Após isso segue-se outra permutação baseada agora nos dois primeiros bits de marcação do terceiro registrador gerando $\{13, 8, 7, 2\}$. Como o terceiro bit a ser testado é igual a 1, o teste marca apenas o número $\{13\}$.

D. Desempenho do Algoritmo

A principal questão neste algoritmo é relacionada a quantidade de bits de b_i necessários para a filtragem de um único elemento do grupo. No caso do segundo exemplo, quando a palavra é $b = 101$, apenas três iterações são necessárias, mas se caso o terceiro bit fosse zero, o algoritmo necessitaria de mais algumas iterações. A hipótese levada em consideração neste trabalho é a de que o algoritmo de Blum-Micali utiliza-se de grupos e geradores que distribuem o melhor possível os elementos quando estes são permutados. Desta forma a filtragem reduzirá, em média, o número de elementos pela metade a cada passo. Esta estimativa define uma curva exponencial de redução, exigindo uma quantidade de iterações de filtragem da ordem de $\mathcal{O}(\log p)$ passos.

V. CONCLUSÕES

O circuito proposto nesse artigo realiza um ataque ao gerador BM em $\log p\sqrt{\log p}$ passos. Duas linhas de investigação estão sendo seguidas para a melhoria dos resultados obtidos. A primeira está relacionada a uma melhor análise da estrutura do grupo finito que define a estrutura de permutação e filtragem utilizada pela solução proposta. Os autores estão analisando os resultados apresentados em [11] a fim de melhorar a estimativa da quantidade de bits de b necessários a uma filtragem perfeita.

Uma segunda linha relaciona-se a real eliminação dos termos da superposição que não satisfazem o bit b_i antes da etapa de permutação. O fato de eliminar esses termos anula a necessidade da utilização do algoritmo de Grover no final da execução. Nessa linha, os autores estão analisando a possibilidade de utilizar um resultado proposto em [6], o lema da amplificação. Esse lema estabelece que sob determinadas condições, é possível para um circuito Q indicado na Figura 3, construir um circuito R que obtenha a saída $|0\rangle \otimes |\phi_0\rangle$ no circuito Q em no máximo $(1/p)\log 1/\varepsilon$ iterações com probabilidade de no mínimo $1 - \varepsilon$. Com isso, poderia-se por exemplo eliminar com alta probabilidade todos os termos da superposição que possuem o primeiro qubit $|1\rangle$ quando o bit b_i correspondente fosse igual a 0.

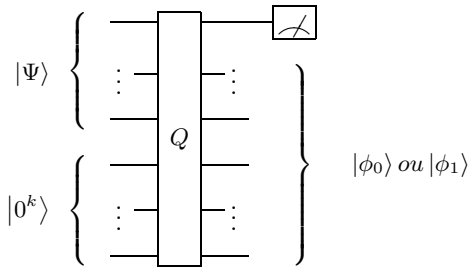


Fig. 3. O circuito Q possui como entrada um estado $|\psi\rangle \otimes |0^{\otimes k}\rangle$. A sua saída consiste em $|0\rangle \otimes |\phi_0\rangle$ ou $|1\rangle \otimes |\phi_1\rangle$.

AGRADECIMENTOS

Os autores agradecem aos órgãos de fomento CAPES e CNPq pelo suporte financeiro ao projeto IQuanta(CT-INFRA/FINEP 01.04.0061.00). Agradecem também aos membros do IQuanta Bernardo Lula, Aércio de Lima e Cheyenne Ribeiro pelas valiosas discussões acerca desse trabalho.

REFERÊNCIAS

[1] W. Stallings, *Cryptography and Network Security: Principles and Practice*. Prentice-Hall, Inc., New Jersey, 2 edition, 1998.

[2] Michael A. Nielsen and Isaac L. Chuang, *Quantum Computation and Quantum Information*. Cambridge University Press, 2000.

[3] Dirk Bouwmeester, Artur Ekert, and Anton Zeilinger, *The Physics of Quantum Information*. Springer, 2000.

[4] Leslie E. Ballentine, *Quantum Mechanics - A Modern Development*. World Scientific, 1998.

[5] R. A. C. Medeiros, *Protocolo para Autenticação Quântica de Mensagens Clássicas*. Dissertação de Mestrado, UFCG, 2004.

[6] John Watrous, *Zero-Knowledge Against Quantum Attacks*. Proceedings of the 38th ACM Symposium on Theory of Computing (STOC'06), 296–305, USA, 2006.

[7] Andrey Sidorenko and Berry Schoenmakers, *State Recovery Attacks on Pseudorandom Generators*. Proceedings of the Western European Workshop on Research in Cryptology (WEWoRC 2005), 53–64, Leuven-Heverlee, Belgium, 2005.

[8] Lov K. Grover, *A fast quantum mechanical algorithm for database search*. Proceedings of 28th ACM Symposium on Theory of Computing (STOC'1996), 212–219, 1996.

[9] Michel Boyer, Gilles Brassard, Peter Hoeyer, and Alain Tapp, *Tight bounds on quantum searching*. In Fortschritte der Physik, v. 46, p. 493, 1998.

[10] Charles Bennett, *Logical Reversibility of Computation*. In IBM Journal of Research and Development, v. 17, p. 525, 1973.

[11] Daniel R. Cloutier, *Mapping the Discrete Logarithm*. Dissertação de Mestrado, Rose-Hulman Institute of Techonology, 2005.