Examples of the Generalized Quantum Permanent Compromise Attack to the Blum-Micali Construction

Elloá B. Guedes, Francisco M. de Assis, Bernardo Lula Jr.
IQuanta – Institute for Studies in Quantum Computation and Quantum Information
Federal University of Campina Grande
Rua Aprígio Veloso, 882 – Campina Grande – Paraíba – Brazil
elloaguedes@gmail.com, fmarcos@dee.ufcg.edu.br, lula@dsc.ufcg.edu.br

September 6, 2010

File Description

This file contains examples of the generalized quantum permanent compromise attack to the Blum-Micali construction. The examples presented here illustrate the attack described in the paper published by Guedes et al. in WECIQ 2010 [3].

To characterize the Blum-Blum-Shub generator, the following references were used: [8, 10, 5, 1]. In the case of the Kaliski generator, the references were: [6, 8, 2, 10]. The reader should consulte them to see more details about these generators.

1 Blum-Blum-Shub Generator

Let M be the product of two large primes p and q where $p \equiv q \equiv 3 \bmod 4$, i.e., M is a Blum prime. Define QR_M as the quadratic residues modulo M, i.e., $QR_M = (\mathbb{Z}_M^*)^2$.

Let $f : \mathbb{Z}_M \to \mathbb{Z}_M$ be the Rabin function, with the following definition

$$f(x) = x^2 \bmod M \tag{1}$$

The Blum-Blum-Shub generator (BBS) takes $x_0 \in_R \mathbb{Z}_M^*$ and iterates the Rabin function in the following way:

$$x_i = x_{i-1}^2 \bmod M \tag{2}$$
$$b_i = \gamma_j(x_i) \tag{3}$$

where γ_j denotes the hard-core predicate for the one-way permutation. This hard-core predicate returns the j-th bit from the given parameter, where j is previously fixed and $1 < j < n$. The value of M and j are publicly know and the security of the BBS generator relies on the hypothesis of the hardness of factoring [5, 8, 10].

Suppose that a cryptosystem uses the BBS to produce pseudorandom quantities. This generator was initialized with the parameters ($M = 3{\cdot}7 = 21, j = 5$) that are publicly known[1].

Suppose that an adversary of this cryptosystem wants to attack the BBS generator. In this scenario, suppose that the adversary (i) discovered that the following sequence of bits $\mathbf{b} = 10$ was outputted by the generator; and, (ii) possess a quantum computer able to execute the generalized quantum permanent compromise attack to the Blum-Micali construction.

In the next sections, the activities to perform the attack successfully will be described.

[1] Considering $j = 5$ represents that the least significant bit will be returned by the hard-core predicate.

1.1 Attack Setup

The attack setup comprehend all the steps necessary to prepare the quantum algorithm to run. Firstly,the adversary needs to prepare the quantum gates that will be used in the attack.

The number of qubits to represent the domain in a quantum computer is $\lceil \log \mathcal{D} \rceil = 5$. Since 2 bits where discovered by the adversary, 2 qubits will compose the second register. In this way, the summarization of necessary qubits is: 5 qubits to first register, 2 qubits to the second register, and 1 qubit as ancillary to the amplitude amplification procedure.

The ρ gate implements the permutation over QR_M, that performs the following transformations:

$$|x \in QR_M\rangle \quad \rightarrow \quad \left|x^2 \bmod M\right\rangle \tag{4}$$

$$|x \notin QR_M\rangle \quad \rightarrow \quad |x\rangle \tag{5}$$

To facilitate the notation, let $lsb(x)$ be the function that, given an integer x, returns the least significant bit of x.

The δ_{b_i} gates, where b_i represents the associated bit produced, have the following definition:

$$\delta_{b_i} |x\rangle |y\rangle = \begin{cases} |x\rangle |\overline{y}\rangle & \text{if } lsb(x) = b_i \text{ and } x \in QR_M \\ |x\rangle |y\rangle & \text{otherwise} \end{cases}$$

In summary, it can be said that the gate δ_{b_i} inverts the target qubit, when the value of the control qubit would have produced the associated bit b_i according to the hard-core predicate lsb.

The last step of the attack setup is to determine how many Grover's iterations will be necessary. In this case, it is expected just a single solution over $N = \lceil \log M \rceil = 5$ bits of input, i.e., 32 numbers. So, the number of iterations k is given by:

$$k = \left\lfloor \frac{\pi}{4} \sqrt{\frac{32}{1}} \right\rceil = 4 \tag{6}$$

Arranging the gates as suggested by the algorithm, the resulting circuit is denoted in the Figure 1.

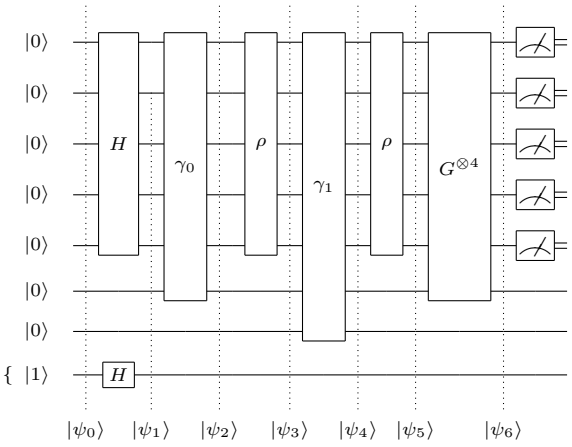

Figure 1: Quantum circuit that implements the attack against the BBS generator.

1.2 Attack Example

Since the requirements for the attack are prepared, the generalized quantum permanent compromise attack is ready to be executed.

The first step is to prepare the four input registers, as shown in $|\psi_0\rangle$ below:

$$|\psi_0\rangle = |00000\rangle |00\rangle |1\rangle \tag{7}$$

A superposition of the input is made to represent all the domain of the generator. The last qubit is also put in superposition because it will be used in the amplitude amplification phase:

$$|\psi_1\rangle = \frac{1}{\sqrt{32}} \left(\sum_{i=0}^{31} |i\rangle \right) |00\rangle |-\rangle \tag{8}$$

Emphasizing the domain QR_M, the state $|\psi_1\rangle$ can be rewritten as:

$$
\begin{aligned}
|\psi_1'\rangle &= \frac{1}{\sqrt{32}}\left(\sum_{i=0}^{31}|i\rangle\right)|00\rangle\,|-\rangle \quad (9)\\
&= \frac{1}{\sqrt{32}}\left(|1\rangle+|4\rangle+|7\rangle+|9\rangle+|15\rangle+\right.\\
&+ \left.|16\rangle+|18\rangle\right)|00\rangle\,|-\rangle+\\
&+ \frac{1}{\sqrt{32}}\sum_{i=0,i\notin QR_M}^{31}|i\rangle\,|00\rangle\,|-\rangle \quad (10)
\end{aligned}
$$

With the first observed bit $b_1 = 1$, the δ_1 gate will be applied, resulting:

$$
\begin{aligned}
|\psi_2\rangle &= \gamma_1\,|\psi_1\rangle \quad (11)\\
&= \frac{1}{\sqrt{32}}\left(|1\rangle+|7\rangle+|9\rangle+|15\rangle\right)|10\rangle\,|-\rangle+\\
&+ \frac{1}{\sqrt{32}}\left(|4\rangle+|16\rangle+|18\rangle\right)|00\rangle\,|-\rangle+\\
&+ \frac{1}{\sqrt{32}}\sum_{i=0,i\notin QR_M}^{31}|i\rangle\,|00\rangle\,|-\rangle \quad (12)
\end{aligned}
$$

Up to this point, the algorithm identify $\hat{X}_1 = \{1, 7, 9, 15\}$ as the potential candidates to the representative. It is important to notice that this identification is just in the quantum level.

The Rabin function, implemented by the ρ gate, must be applied to the input:

$$
\begin{aligned}
|\psi_3\rangle &= \rho\,|\psi_2\rangle \quad (13)\\
&= \frac{1}{\sqrt{32}}\left(|1\rangle+|7\rangle+|18\rangle+|15\rangle\right)|10\rangle\,|-\rangle+\\
&+ \frac{1}{\sqrt{32}}\left(|4\rangle+|16\rangle+|9\rangle\right)|00\rangle\,|-\rangle+\\
&+ \frac{1}{\sqrt{32}}\sum_{i=0,i\notin QR_M}^{31}|i\rangle\,|00\rangle\,|-\rangle \quad (14)
\end{aligned}
$$

The second bit will be used to determine $\hat{X}_2$:

$$
\begin{aligned}
|\psi_4\rangle &= \gamma_0\,|\psi_3\rangle \quad (15)\\
&= \frac{1}{\sqrt{32}}\,|18\rangle\,|11\rangle\,|-\rangle+\\
&+ \frac{1}{\sqrt{32}}\left(|1\rangle+|7\rangle+|15\rangle\right)|10\rangle\,|-\rangle+\\
&+ \frac{1}{\sqrt{32}}\left(|4\rangle+|16\rangle\right)|01\rangle\,|-\rangle+|9\rangle\,|00\rangle\,|-\rangle+\\
&+ \frac{1}{\sqrt{32}}\sum_{i=0,i\notin QR_M}^{31}|i\rangle\,|00\rangle\,|-\rangle \quad (16)
\end{aligned}
$$

It is important to notice that $\hat{X}_2 = \{9\}$ and the solution is already identified in a quantum level. The next step is to simply obtain x_3:

$$
\begin{aligned}
|\psi_5\rangle &= \rho\,|\psi_4\rangle \quad (17)\\
&= \frac{1}{\sqrt{32}}\,|9\rangle\,|11\rangle\,|-\rangle+\\
&+ \frac{1}{\sqrt{32}}\left(|1\rangle+|7\rangle+|15\rangle\right)|10\rangle\,|-\rangle+\\
&+ \frac{1}{\sqrt{32}}\left(|16\rangle+|4\rangle\right)|01\rangle\,|-\rangle+|18\rangle\,|00\rangle\,|-\rangle\\
&+ +\frac{1}{\sqrt{32}}\sum_{i=0,i\notin QR_M}^{31}|i\rangle\,|00\rangle\,|-\rangle \quad (18)
\end{aligned}
$$

The state $|\psi_5\rangle$ can be written as a partition, where $z \neq 11$:

$$
\begin{aligned}
|\psi_5'\rangle &= \frac{1}{\sqrt{32}}\,|9\rangle\,|11\rangle\,|-\rangle+\sum_{i=0,i\neq 9}^{31}|i\rangle\,|z\rangle\,|-\rangle \quad (19)\\
&= \frac{1}{\sqrt{32}}\,|\psi_{x_i}\rangle+\sqrt{\frac{31}{32}}\,|\psi_{\neg x_i}\rangle \quad (20)
\end{aligned}
$$

It should be noticed that $|\psi_{x_i}\rangle = |9\rangle\,|11\rangle\,|-\rangle$ and $|\psi_{\neg x_i}\rangle = \sum_{i=0,i\neq 9}^{31}|i\rangle\,|z\rangle\,|-\rangle$.

Considering the geometric representation of this state, then:

$$
|\psi_5'\rangle = \sin\theta\,|\psi_{x_i}\rangle + \cos(\theta)\,|\psi_{\neg x_i}\rangle \quad (21)
$$

where $\sin^2\theta = \frac{1}{32}$ and $\theta \in \left(0, \frac{\pi}{2}\right)$, therefore $\theta = 0.17771$ radians.

The next step is to perform $k = 4$ Grover iterations, resulting:

$$\begin{aligned} |\psi_6\rangle &= G^{\otimes 4}|\psi_5\rangle & (22)\\ &= \sin[(2\cdot k+1)\theta]\,|\psi_{good}\rangle + \\ &+ \cos[(2\cdot k+1)\theta]\,|\psi_{bad}\rangle & (23)\\ &= \sin[9\cdot 0.17771]\,|\psi_{good}\rangle + \\ &+ \cos[9\cdot 0.17771]\,|\psi_{bad}\rangle & (24)\\ &= \sin(1.599)\,|\psi_{good}\rangle + \cos(1.599)\,|\psi_{bad}\rangle & (25)\end{aligned}$$

A measurement in the second register will return 9 with probability of $|\sin(1.599)|^2 \cong 0.9996$. It means that with just two qubits, the representative of the BBS generator was correctly retrieved with high probability.

This concludes an example of the generalized quantum permanent compromise attack against the security of the BBS generator.

2 Kaliski Generator

The Kaliski generator is based on the elliptic curve discrete logarithm problem. Let p be a prime, $p \equiv 2 \bmod 3$, and consider a curve $E(\mathbb{F}_p)$ that consists of points $(x, y) \in \mathbb{F}_p \times \mathbb{F}_p$ such that:

$$y^2 = x^3 + c \tag{26}$$

The points of $E(\mathbb{F}_p)$ together with a point at infinity $\mathcal{O}$ form a cyclic additive group of order $p+1$. Let Q be a generator this group and let ϕ be a function with the following definition:

$$\phi(P) = \begin{cases} y & \text{if } P = (x, y) \\ p & \text{if } P = \mathcal{O} \end{cases}$$

The Kaliski generator's one-way permutation and hard-core predicate are given below:

$$\begin{aligned} f(P) &= \phi(P)Q & (27)\\ b_i &= \lambda(P) & (28)\end{aligned}$$

where the function λ has the following definition:

$$\lambda(P) = \begin{cases} 1 & \text{if } \phi(P) \geq \frac{p+1}{2} \\ 0 & \text{otherwise} \end{cases}$$

The domain of the Kaliski generator is $\mathcal{D} = E(\mathbb{F}_p)$ and the seed P_1 is a random point on the curve.

Suppose that a cryptosystem uses the Kaliski generator to produce pseudorandom quantities. This generator was initialized with the parameters $p = 5$ and $c = 1$. Suppose also that an adversary of this cryptosystem wants to attack a Kaliski generator.

In this scenario, suppose that the adversary (i) discovered that the following sequence of bits $\mathbf{b} = 10$ was outputted by the generator; and, (ii) possess a quantum computer able to execute the generalized quantum permanent compromise attack to the Blum-Micali construction.

In the next section, details about the Kaliski generator under attack will be presented to the reader in order to clarify the comprehension about the steps of the attack. After that, the attack setup will be described, reporting all the gates and number of iterations required by the attack. To conclude the attack, the steps of the quantum algorithm will be detailed.

3 Details of Initialization of the Kaliski Generator Under Attack

In the example of the Kaliski generator used in this file, the initialization adopted the parameters $p = 5$ and $c = 1$, resulting in the following equation of the curve:

$$y^2 = x^3 + 1 \bmod 5 \tag{29}$$

The set of points that satisfy this equation is $\{(4,0),(0,1),(0,4),(2,2),(2,3)\}$. This set together with a point at infinity, denoted by $\mathcal{O}$, characterizes the cyclic group of order $p+1$, i.e., the domain of the permutation.

The generator of this group is $Q = (2, 2)$ and is important to remark that:

$$
\begin{aligned}
Q &= (2,2) && (30)\\
2Q &= Q+Q=(0,4) && (31)\\
3Q &= 2Q+Q=(4,0) && (32)\\
4Q &= 3Q+Q=(0,1) && (33)\\
5Q &= 4Q+Q=(2,3) && (34)\\
6Q &= 5Q+Q=\mathcal{O} && (35)
\end{aligned}
$$

It is important to notice that kQ, where k is an integer, does not represent the ordinary multiplication operation. It represents the addition of a point to itself in the context of an elliptic curve. More details about this operation should be seen in the book of Paar and Pelzl (Section 9.1.2 – Group Operations on Elliptic Curves) [7] and also in the book of Stallings (Section 6.5 – Elliptic Curves Over Finite Fields) [9].

The generator of the example has the form:

$$
\begin{aligned}
P_i &= \phi(P_{i-1})Q && (36)\\
b(P_i) &= \lambda(P) && (37)
\end{aligned}
$$

where the function ϕ has the following definition:

$$
\phi(P) = \begin{cases} y & \text{if } P=(x,y) \\ p & \text{if } P=\mathcal{O} \end{cases}
$$

The function λ has the following definition:

$$
\lambda(P) = \begin{cases} 1 & \text{if } \phi(P) \geq 3 \\ 0 & \text{otherwise} \end{cases}
$$

For this example, the resulting permutation can be represented as the functional graph illustrated in the Figure 2.

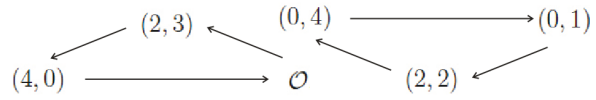

Figure 2: Functional graph for the one-way permutation of the Kaliski generator used in the example.

3.1 Attack Setup

The attack setup comprehend all the steps necessary to prepare the quantum algorithm to run. Firstly is is necessary to determine how many qubits are necessary as input.

The number of qubits to represent the domain in a quantum computer is $\lceil \log \mathcal{D} \rceil = \lceil \log 6 \rceil = 3$. Since 2 bits where discovered by the adversary, 2 qubits will be necessary in the third register. In this way, the summarization of necessary qubits is: 3 qubits to first register, 2 qubits to the second register, and 1 qubit as ancillary to the amplitude amplification procedure.

Since the points cannot be directly represented in a quantum computer, the following representation will be used:

$$
\begin{aligned}
(4,0) &\equiv |1\rangle && (38)\\
(0,1) &\equiv |2\rangle && (39)\\
(0,4) &\equiv |3\rangle && (40)\\
(2,2) &\equiv |4\rangle && (41)\\
(2,3) &\equiv |5\rangle && (42)\\
\mathcal{O} &\equiv |6\rangle && (43)
\end{aligned}
$$

The next step is to to prepare the quantum gates that will be used in the attack. The ρ gate, responsible to implement the permutation, performs the following transformations:

$$
\begin{aligned}
|0\rangle &\rightarrow |0\rangle && (44)\\
|1\rangle &\rightarrow |6\rangle && (45)\\
|2\rangle &\rightarrow |4\rangle && (46)\\
|3\rangle &\rightarrow |2\rangle && (47)\\
|4\rangle &\rightarrow |3\rangle && (48)\\
|5\rangle &\rightarrow |1\rangle && (49)\\
|6\rangle &\rightarrow |5\rangle && (50)\\
|7\rangle &\rightarrow |7\rangle && (51)
\end{aligned}
$$

It should be noticed that the gate ρ is unitary, since $\rho \cdot \rho^{\dagger} = \mathbb{I}$, where $\mathbb{I}$ denotes the identity matrix.

The gate λ_0 performs the following transformations:

$$|0\rangle|c\rangle \rightarrow |0\rangle|c\rangle \quad (52)$$
$$|1\rangle|c\rangle \rightarrow |1\rangle|\overline{c}\rangle \quad (53)$$
$$|2\rangle|c\rangle \rightarrow |2\rangle|\overline{c}\rangle \quad (54)$$
$$|3\rangle|c\rangle \rightarrow |3\rangle|c\rangle \quad (55)$$
$$|4\rangle|c\rangle \rightarrow |4\rangle|\overline{c}\rangle \quad (56)$$
$$|5\rangle|c\rangle \rightarrow |5\rangle|c\rangle \quad (57)$$
$$|6\rangle|c\rangle \rightarrow |6\rangle|c\rangle \quad (58)$$
$$|7\rangle|c\rangle \rightarrow |7\rangle|c\rangle \quad (59)$$
$$(60)$$

In the case of the Kaliski generator, the matrix representation of the gates is shown in the Appendix A. The reader can verify that they are unitary by performing a multiplication of each gate to it transpose conjugated.

The number of iterations required by the Grover's algorithm is given by:

$$k = \left\lfloor \frac{\pi}{4}\sqrt{\frac{8}{1}} \right\rfloor = 2 \quad (61)$$

Arranging the gates as suggested by the algorithm, the resulting circuit is denoted in the Figure 3.

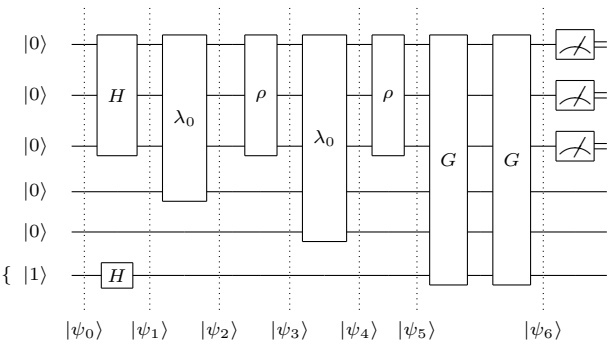

Figure 3: Quantum circuit that implements the attack against the Kaliski generator.

3.2 Attack Example

The first step describes the initialization of the circuit according to each register as shown in the $|\psi_0\rangle$:

$$|\psi_0\rangle = |000\rangle|00\rangle|1\rangle \quad (62)$$

It is applied to the first and third registers the Hadamard gate, responsible to put the input in an equally distributed superposition. The result of the application of such gate is shown in the $|\psi_1\rangle$:

$$|\psi_1\rangle = H^{\otimes 3} \otimes I^{\otimes 2} \otimes H\,|\psi_0\rangle \quad (63)$$
$$= H^{\otimes 3}\,|000\rangle|00\rangle\,H\,|1\rangle \quad (64)$$
$$= \frac{1}{\sqrt{8}}\sum_{i=0}^{8}|i\rangle|00\rangle|-\rangle \quad (65)$$
$$= \frac{1}{\sqrt{8}}\left(|0\rangle + |1\rangle + \ldots |7\rangle\right)|00\rangle|-\rangle \quad (66)$$

At this point, all the states have the same probability to be measured. The next step is to perform the first phase of the quantum permanent compromise algorithm, responsible for the identification of the representative. The λ_0 gate associate in the third register all the elements of the first one that would have produced the bit 0 in the hard-core predicate. The result is shown in the $|\psi_2\rangle$ below:

$$|\psi_2\rangle = \lambda_0\,|\psi_1\rangle \quad (67)$$
$$= \frac{1}{\sqrt{8}}\left(|1\rangle + |2\rangle + |4\rangle\right)|10\rangle|-\rangle + $$
$$+ \frac{1}{\sqrt{8}}\left(|0\rangle + |3\rangle + |5\rangle + |6\rangle + |7\rangle\right)|00\rangle|-\rangle \quad (68)$$

It is important to notice that up to this point the the candidates to the representative are: $\{|1\rangle, |2\rangle, |4\rangle\}$. Since the algorithm reproduces the steps of the Kaliski generator, it is necessary to perform the permutation in all the elements of the domain. This operation is performed by the ρ gate, as shown in the state $|\psi_3\rangle$.

$$|\psi_3\rangle = \rho\,|\psi_2\rangle \quad (69)$$
$$= \frac{1}{\sqrt{8}}\left(|6\rangle + |4\rangle + |3\rangle\right)|10\rangle|-\rangle + $$
$$+ \frac{1}{\sqrt{8}}\left(|0\rangle + |2\rangle + |1\rangle + |5\rangle + |7\rangle\right)|00\rangle|-\rangle \quad (70)$$

The next step is to apply again the gate λ_0, that will identify the elements that would have produced

the second bit. The effect of this gate is reported in the $|\psi_4\rangle$.

$$\begin{aligned} |\psi_4\rangle &= \lambda_0 |\psi_3\rangle & (71)\\ &= \frac{1}{\sqrt{8}}|4\rangle|11\rangle|-\rangle + \frac{1}{\sqrt{8}}(|2\rangle+|1\rangle)|01\rangle|-\rangle + \\ &+ \frac{1}{\sqrt{8}}(|6\rangle+|3\rangle)|10\rangle|-\rangle + \\ &+ \frac{1}{\sqrt{8}}(|0\rangle+|5\rangle+|7\rangle)|00\rangle|-\rangle & (72)\end{aligned}$$

The next step is to perform the application of the gate ρ one more time. It is necessary to identify the representant of the internal state $X(3)$.

$$\begin{aligned} |\psi_5\rangle &= \rho|\psi_4\rangle & (73)\\ &= \frac{1}{\sqrt{8}}|3\rangle|11\rangle|-\rangle + \frac{1}{\sqrt{8}}(|4\rangle+|6\rangle)|01\rangle|-\rangle \\ &+ \frac{1}{\sqrt{8}}(|5\rangle+|4\rangle)|10\rangle|-\rangle + \\ &+ \frac{1}{\sqrt{8}}(|0\rangle+|1\rangle+|7\rangle)|00\rangle|-\rangle & (74)\end{aligned}$$

After that, it is important to notice that the representative of the internal state $X(3)$ is already identified: $|3\rangle$. However, a measurement in the second register at this point would return any number from $|0\rangle$ to $|7\rangle$ with the same probability. The next step of the algorithm comprehend the amplitude amplification of the element identified as solution. To proceed is necessary to consider the following representation of the state $|\psi_5\rangle$:

$$\begin{aligned} |\psi_5'\rangle &= \frac{1}{\sqrt{8}}|3\rangle|11\rangle|-\rangle + \frac{1}{\sqrt{8}}\sum_{j=0,j\neq 3}^{7}|j\rangle|z\neq 11\rangle & (75)\\ &= \frac{1}{\sqrt{8}}|\psi_{x_i}\rangle + \sqrt{\frac{7}{8}}|\psi_{\neg x_i}\rangle & (76)\end{aligned}$$

It should be noticed that there's a partition in two subspaces:$|\psi_{x_i}\rangle = |3\rangle|11\rangle|-\rangle$ and $|\psi_{\neg x_i}\rangle = \sum_{j=0,j\neq 3}^{7}|j\rangle|z\neq 11\rangle|-\rangle$.

Considering the geometric representation of this state, then:

$$|\psi_5'\rangle = \sin\theta|\psi_{x_i}\rangle + \cos(\theta)|\psi_{\neg x_i}\rangle \quad (77)$$

where $\sin^2\theta = \frac{1}{8}$ and $\theta \in \left(0, \frac{\pi}{2}\right)$, therefore $\theta = 0.361$ radians.

The next step of the algorithm is to perform $k = 2$ Grover's iterations in the state $|\psi_5'\rangle$, resulting:

$$\begin{aligned} |\psi_6\rangle &= G^{\otimes 2}|\psi_5'\rangle & (78)\\ &= sin[(2\cdot k+1)\theta]|\psi_{x_i}\rangle + \\ &+ \cos[(2\cdot k+1)\theta]|\psi_{\neg x_i}\rangle & (79)\\ &= \sin[5\cdot 0.361]|\psi_{good}\rangle + \\ &+ \cos[5\cdot 0.361]|\psi_{bad}\rangle & (80)\\ &= \sin(1.805)|\psi_{good}\rangle + \cos(1.805)|\psi_{bad}\rangle & (81)\end{aligned}$$

At this point, a measurement in the second register would return the state $|3\rangle$ with probability of $|\sin(1.805)|^2 = 0.946$. With this information the intruder will be able to retrieve all the set $X(i)$ of internal states from the generator under attack, endangering its unpredictability.

This concludes an example of the generalized quantum permanent compromise attack against the security of the Kaliski generator.

4 Final Remarks

The examples illustrated in this file show how to endanger the security of the generators BBS and Kaliski from the Blum-Micali Construction. This endangering is made by a quantum permanent compromise attack and the consequence is that an adversary is capable to reproduce all the previous and future outputs of the generator.

The quantum attack is based on Amplitude Amplification, a generalization of Grover's quantum search. This attack provides a quadratic speedup over the classical analogous algorithm. For more details about the quantum attack, the reader is reported to the papers of Guedes et al. [4, 3].

Acknowledgements

The authors gratefully acknowledge the financial support rendered by the Brazilian National Council for the Improvement of Higher Education (CAPES).

References

[1] L. Blum, M. Blum, and M. Shub. A Simple Unpredictable pseudo-random number generator. *SIAM Journal on Computing*, pages 364–383, 1986.

[2] Afonso Comba de Araújo Neto. Um algoritmo de criptografia de chave pública semanticamente seguro baseado em curvas elípticas. Master's thesis, Universidade Federal do Rio Grande do Sul, 2006.

[3] Elloá B. Guedes, Francisco M. de Assis, and Bernardo Lula Jr. A Generalized Quantum Permanent Compromise Attack to the Blum-Micali Construction. In *Workshop-Escola de Computação e Informação Quânticas*, 2010.

[4] Elloá B. Guedes, Francisco M. de Assis, and Bernardo Lula Jr. Quantum Permanent Compromise Attack to Blum-Micali Pseudorandom Generator. In *Proceedings of the IEEE International Telecommunications Symposium*, 2010.

[5] Pascal Junod. Cryptographic Secure Pseudo-Random Bits Generation: The Blum-Blum-Shub Generator, 1999. http://crypto.junod.info/publications/.

[6] B. S. Kaliski. *Elliptic Curves and Cryptography: A Pseudorandom Bit Generator and Other Tools*. PhD thesis, MIT, Cambridge, MA, USA, 1988.

[7] Christof Paar and Jan Pelzl. *Understanding Cryptography*. Springer, 2010.

[8] Andrey Sidorenko and Berry Schoenmakers. State Recovery Attacks on Pseudorandom Generators. In *Western European Workshop on Research in Cryptology*, pages 53–63, 2005.

[9] William Stallings. *Crytpgraphy and Network Security*. Prentice Hall, 1999.

[10] Henk C.A. van Tilborg. *Encyclopedia Of Cryptography and Security*. Springer, 2005.

A Matrix Representation of the Gates

$$\rho = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\lambda_0 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$