On the Use of Entanglement: Part 2

Du Linglong

Donghua University

matdl@dhu.edu.cn

January 6, 2021

Overview

Overview

Preparation: Group theory(Ref. Appendix F)

Definition

Definition 6.15 Let $\mathcal{H}$ be a subgroup of the group $\mathcal{G}$ and let S be a finite set. We say that a function $f : \mathcal{G} \to S$ **hides the subgroup** $\mathcal{H}$ if for any $g_1, g_2 \in \mathcal{G}$

$$f(g_1) = f(g_2) \Leftrightarrow g_1^{-1} g_2 \in \mathcal{H}.$$

ex:6.73

Remark

f hides $\mathcal{H}$ $\Leftrightarrow$ it is constant on any given left coset and takes different values on distinct left cosets of $\mathcal{H}$, i.e.,

$$\forall g_1, g_2 \in \mathcal{G}, \quad f(g_1) = f(g_2) \Leftrightarrow g_1\mathcal{H} = g_2\mathcal{H}.$$

Abelian Hidden Subgroup Problem algorithm

Definition

Definition 6.16 Let f hide the subgroup $\mathcal{H}$ of the group $\mathcal{G}$. The problem to identify $\mathcal{H}$ with the help of f is called **Hidden Subgroup Problem(HSP)**. In case $\mathcal{G}$ is a finite abelian group it is called the **Abelian Hidden Subgroup Problem (AHSP)**.

- In the following, we consider the case: $|\mathcal{G}|$ is finite.

Under the assumption: $\mathcal{S}_{AHSP\ Step\ i}(|\mathcal{G}|) \in poly(log_2|\mathcal{G}|)$ for $|\mathcal{G}| \to \infty$, $i = 1, 2, 3, 4, 5$.

|S|=m, ordered

- Input: A finite abelian group $\mathcal{G} = \{g_1, \cdots, g_{|\mathcal{G}|}\}$ and a function $f : \mathcal{G} \to S$ that hides a subgroup $\mathcal{H} \leq \mathcal{G}$.
- Step 1: Prepare the initial state $|\Psi_0\rangle = \frac{1}{\sqrt{|\mathcal{G}|}} \sum_{g \in \mathcal{G}} |g\rangle \otimes |0\rangle \in \mathbb{H}^A \otimes \mathbb{H}^B$,

ONB

$$\mathbb{H}^A = Span\{|g_1\rangle, |\cdots, g_{|\mathcal{G}|}\rangle\} \subset {}^{\P}\mathbb{H}^{\otimes n}, \quad n = \lceil \log_2 |\mathcal{G}| \rceil,$$

$$\mathbb{H}^B = {}^{\P}\mathbb{H}^{\otimes m}, \quad m = \lceil \log_2 |S| \rceil.$$

Algorithm

- Step 2: Apply U_f to $|\Psi_0\rangle$ to produce

$$|\Psi_1\rangle = U_f|\Psi_0\rangle = \frac{1}{\sqrt{|\mathcal{G}|}} \sum_{g\in\mathcal{G}} |g\rangle \otimes |\widetilde{f(g)}\rangle \in \mathbb{H}^A \otimes \mathbb{H}^B,$$

unitary operator

$$U_f : \mathbb{H}^A \otimes \mathbb{H}^B \to \mathbb{H}^A \otimes \mathbb{H}^B$$

$$|g\rangle \otimes |y\rangle \longmapsto |g\rangle \otimes y \boxplus \widetilde{f(g)}\rangle.$$

ONB

Algorithm

In the follow steps

- Consider only sub-system $\mathbb{H}^A$, described by the mixed state

$$\rho^A = tr^B(\rho) = tr^B(|\Psi_1\rangle\langle|\Psi_1) = \frac{|\mathcal{H}|}{|\mathcal{G}|}\sum_{[g]_\mathcal{H}|\in\mathcal{G}/\mathcal{H}} |\Psi^A_{[g]_\mathcal{H}}\rangle\langle\Psi^A_{[g]_\mathcal{H}}|,$$

quotient group

$$|\Psi^A_{[g]_\mathcal{H}}\rangle := \frac{1}{\sqrt{|\mathcal{H}|}}\sum_{h\in[g]_\mathcal{H}} (|h\rangle.$$

- Define $F_\mathcal{G} = \frac{1}{\sqrt{|\mathcal{G}|}}\sum_{g\in\mathcal{G}}\sum_{\chi\in\hat{\mathcal{G}}}\chi(g)|\chi\rangle\langle g|$

the dual group

- Define $\mathcal{H}^\perp := \{\chi\in\hat{\mathcal{G}}|\mathcal{H}\subset Ker(\chi)\}$

Lemma

Lemma F.39 Let $\mathcal{H}$ be the subgroup $\mathcal{G}$. Then $\mathcal{H}^\perp$ is a subgroup of $\hat{\mathcal{G}}$.

Algorithm

- Step 3: Perform the quantum FOURIER transform $F_{\mathcal{G}}$ to transform sub-system $\mathbb{H}^A$ into the state

$$F_{\mathcal{G}} \rho^A F_{\mathcal{G}}^* = \sqrt{\frac{|\mathcal{H}|}{|\mathcal{G}|}} \sum_{[g]_{\mathcal{H}}| \in \mathcal{G}/\mathcal{H}} \left(\sum_{\chi \in \mathcal{H}^{\perp}} \chi(g) |\chi\rangle \right) \left(\sum_{\xi \in \mathcal{H}^{\perp}} \overline{\xi(g)} \langle \xi | \right),$$

i.e., $\rho^A \longmapsto F_{\mathcal{G}} \rho^A F_{\mathcal{G}}^*$.

Algorithm

- Step 4: Observe the sub-system $\mathbb{H}^A$ to detect a $\xi \in \mathcal{H}^\perp$ with certainty.
- Step 5: Repeat Steps 1-4 for $L \geq \log_2(\frac{|\mathcal{G}|}{\varepsilon|\mathcal{H}|})$ times to determine $\xi_1, \cdots, \xi_L \in \mathcal{H}^\perp$ and form $\bigcap_{l=1}^{L} Ker(\xi_l)$.
- Output: The desired $\mathcal{H} = \bigcap_{l=1}^{L} Ker(\xi_l)$, with a probability $P\{\langle \xi_i, \cdots, \xi_L \rangle = \mathcal{H}^\perp\} \geq 1 - \varepsilon$.

Corollary F.50

$\varepsilon = \frac{|G|}{2^L}$

Definition

Play an essential role in some advanced cryptographic protocols

Definition

Definition 6.17 Let $\mathcal{G}$ be a group and $g, h \in \mathcal{G}$ such that there exists a $d \in \mathbb{N}_0$ such that $h = g^d$.
Then d is called the **discrete logarithm** of h to base g, and this is expressed by the notation $d = \mathrm{dlog}_g(h)$.
The task to find $d = \mathrm{dlog}_g(h)$, when only g and h are known, is called the **Discrete Logarithm Problem(DLP)**.

Remark

The **DLP** *can be formulated as an AHSP, for a suitably chosen group, set and function in the* **AHSP***.*

Algorithm

unit element

- Given:
  (i) A group $\mathcal{G}_{DLP}$ and a element $g \in \mathcal{G}_{DLP}$ that has order $N = ord(g)$, that is $N \in \mathbb{N}$ is the smallest number satisfying $g^N = e_{\mathcal{G}_{DLP}}$.
  (ii) An $h \in \mathcal{G}_{DLP}$ such that $h = g^d$, for some unknown $d \in \mathbb{N}$.
- Aim: Find $d = \mathrm{dlog}_g(h)$.
- We choose: $\mathcal{G} := \mathbb{Z}_N \times \mathbb{Z}_N$,
  $g = ([x]_{N\mathbb{Z}}, [y]_{N\mathbb{Z}}) = (x \mathrm{mod} N, y \mathrm{mod} N) \in \mathcal{G}$.
  $S := \langle g \rangle \leq \mathcal{G}_{DLP}$.

cyclic group,|S|= N

Algorithm

- Step 1: Initial preparation:

$$\mathbb{H}^A = Span\{|u\rangle \otimes |v\rangle | u, v \in \{0, \cdots, N-1\}\},$$

choose

$$f : \mathcal{G} \to S$$
$$([x]_{N\mathbb{Z}}, [y]_{N\mathbb{Z}}) \longmapsto h^x g^y,$$

i.e., $f([x]_{N\mathbb{Z}}, [y]_{N\mathbb{Z}}) = (g^d)^x g^y = g^{dx+y} \in \langle g \rangle$.
$\mathcal{H} = \{([u]_{N\mathbb{Z}}, [-du]_{N\mathbb{Z}}) | [u]_{N\mathbb{Z}} \in \mathbb{Z}_N\} \leq \mathcal{G}$.

ex 6.75

Algorithm

- Step 2: Fourier transform $F_{\mathcal{G}}$ to transform sub-system $\mathbb{H}^A$ into $F_{\mathcal{G}}\rho^A F_{\mathcal{G}}^*$, i.e., $\rho^A \longmapsto F_{\mathcal{G}}\rho^A F_{\mathcal{G}}^*$,

$$\rho^A = \frac{|\mathcal{H}|}{|\mathcal{G}|} \sum_{[g]_{\mathcal{H}}|\in \mathcal{G}/\mathcal{H}} |\Psi^A_{[g]_{\mathcal{H}}}\rangle\langle\Psi^A_{[g]_{\mathcal{H}}}|,$$

$$|\Psi^A_{[g]_{\mathcal{H}}}\rangle := \frac{1}{\sqrt{N}} \sum_{[u]_{N\mathbb{Z}}\in\mathbb{Z}_N} |(x+u)\text{mod } N\rangle \otimes |(y-du)\text{mod } N\rangle,$$

$$F_{\mathcal{G}} = \frac{1}{N} \sum_{m,n,v,w\in\{0,\cdots,N-1\}} e^{2\pi i\frac{mv+nw}{N}} |m\rangle\otimes|n\rangle\langle v|\otimes\langle w|$$

- Step 3: Find the state $|\xi\rangle$ corresponds to a character $\xi \in \mathcal{H}^\perp$

$$\mathcal{H}^\perp = \{\chi_{dn\text{mod}N,n} | [n]_{N\mathbb{Z}} \in \mathbb{Z}_N\}$$

- Step 4: Find $\mathcal{H}$ or $d = \text{dlog}_g(h)$
$d = (a(dn\text{mod } N) + b(dm \text{ mod } N))\text{mod } N.$

DSA Protocol

Digital Signature Algorithm (DSA) Protocol

Signer	**Public knows**
	algorithm parameters $\mathcal{A}$
	verification statement v
chooses a private key k	
creates a public ***verification key***	
by	
computing a $V = V(k,\mathcal{A})$	
and publishing it	verification key V
signs document by	
taking document d,	document d
computing a *signature* $s(d,\mathcal{A})$	
and publishing it	signature s
	and can verify by
	checking the verification statement $v(s,d,V,\mathcal{A}) = \text{TRUE}$?

ECDSA Protocol

Elliptic Curve Digital Signature (ECDSA) Protocol

Signer	**Public knows**
	algorithm parameters $\mathcal{A}$:
	large prime p
	elliptic curve $E(\mathbb{F}_p)$
	public point $P \in E(\mathbb{F}_p) \setminus \{0_E\}$
	with a large prime order q
creates key by	
choosing a *secret signing key* $k \in \mathbb{N}$	
with $1 < k < q$,	
computing the *verification key* $V = kP$	
and publishing it	verification key V
signs document by	
taking document d and a random $a \in \mathbb{N}$ with $a < q$,	document d
computing	
$aP \in E(\mathbb{F}_p) \setminus \{0_E\}$	
$s_1 = x_{aP} \bmod q$	
$s_2 = \left((d + s s_1)(a^{-1} \bmod q)\right) \bmod q$	
and publishing the *signature* (s_1, s_2)	signature (s_1, s_2)
	and verifies by
	computing
	$u_1 = \left(d(s_2^{-1} \bmod q)\right) \bmod q$
	$u_2 = \left(s_1(s_2^{-1} \bmod q)\right) \bmod q$
	$(x, y) = u_1 P +_E u_2 V \in E(\mathbb{F}_p) \setminus \{0_E\}$
	and checking the verification statement
	is $x \bmod q = s_1$ **TRUE?**

Public key Generation

Signature Generation

five publicly known parameters (p,A,B,P,q)

Verification

Proposition 6.20

ECDSA Protocol

The security of this depends on the difficulty of the calculation of k.

Remark

The computational steps to calculate $k = dlog_P(V)$ *for the bitcoin ECDSA:*

- *The classical method:* *order of* $O(10^{77})$.
- *The quantum computer:* *order of* $O(polynomial\ in\ \ 256)$. *Thus render the bitcoin signature insecure.*

Brief ideas

> 50% in $O(\sqrt{N})$ steps

- Represent the objects as quantum states, i.e., normalized vectors in a suitable Hilbert space. The vectors of the objects which we try to find span a subspace in this Hilbert space.
- Construct operators that successively transform (or rotate) a given initial state into a state which has a maximal component in the subspace of desired objects.
- Measure the rotated states, with a greater probability of detecting a state which lies in the subspace of desired objects.

Remark

This method of rotating the initial state into the solution space is also used in other quantum algorithms and has become known as **amplification**.

Algorithm

The number of computational steps $S_{Grover}(N) \in O(\sqrt{\frac{N}{m}})$ for $N \to \infty$.

- Input: **A set** $\{0, ..., N-1\}$ of $N = 2^n$ objects
  A subset $S($ solution set$)$ of $m \geq 1$ objects to be searched for
  oracle-function $g : \{0, ..., N-1\} \to \{0, 1\}$ that

$$x \longmapsto g(x) := \begin{cases} 0, \text{ if } & x \in S^{\perp} \\ 1, \text{if } & x \in S \end{cases}$$

Oracle $\hat{U}_g$ via the following action on the computational basis $\hat{U}_g(|x\rangle \otimes |y\rangle) := |x\rangle \otimes |y \boxplus g(x)\rangle$.

Remark

Every number $x \in \{0, \cdots, N-1\}$ *can be uniquely associated with a vector in the computational basis of* $¶\mathbb{H}^{\otimes n}$.

Algorithm

- Step 1: Prepare the composite system in the state $|\hat{\Psi}_0|\rangle = |\Psi_0\rangle \otimes |-\rangle$ in $\mathbb{H}^{I/O} \otimes \mathbb{H}^{W} = ¶\mathbb{H}^{\otimes n} \otimes ¶\mathbb{H}$, $|\Psi_0\rangle = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle$.
- Step2 : Apply the transform $\hat{G} = (R_{\Psi_0} \otimes \mathbf{1})\hat{U}_g \; j_N = \lfloor \frac{\pi}{4\theta_0} \rfloor$ times to $|\hat{\Psi}_0\rangle$ in order to transform the composite system to the state $|\hat{\Psi}_{j_N}\rangle = \hat{G}^{j_N}|\hat{\Psi}_0\rangle$.
- Step3 : Observe the sub-system $\mathbb{H}^{I/O} = ¶\mathbb{H}^{\otimes n}$ and infer from the observed state $|x\rangle$ the value $x \in \{0, \cdots, N-1\}$.
- Step4 : By evaluating $g(x)$, check if $x \in S$.
- Output : A solution $x \in S$ with the probability no less than $1 - \frac{m}{N}$ (Theorem 6.30).

Algorithm

The number of computational steps $S_{Gr\tilde{o}ver}(N) \in O(\sqrt{N})$ for $N \to \infty$.

- Input: A set $\{0, ..., N-1\}$ of $N = 2^n$ objects
  A subset S of $m \geq 0$ objects to be searched for
  An oracle-function $g : \{0, ..., N1\} \to \{0, 1\}$
- Step 1: Randomly select an $x \in \{0, \cdots, N-1\}$ and check if $x \in S$.
  - If TURE: Done.
  - If FALSE: Go to step 2.
- Step 2: Prepare the composite system in the state $|\hat{\Psi}_0|\rangle = |\Psi_0\rangle \otimes |-\rangle$ in $\mathbb{H}^{I/O} \otimes \mathbb{H}^W = ¶\mathbb{H}^{\otimes n} \otimes ¶\mathbb{H}$, $|\Psi_0|\rangle = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle$.

Algorithm

- Step3: Set $J := \lfloor\sqrt{N}\rfloor + 1$ and randomly select an integer $j \in \{0, \cdots, J-1\}$ with equal probability $\frac{1}{J}$. Apply the transform $\hat{G} = (R_{\Psi_0} \otimes \mathbf{1})\hat{U}_g$ j times to $|\hat{\Psi}_0\rangle$ in order to transform the composite system to the state $|\hat{\Psi}_j\rangle = \hat{G}^j|\hat{\Psi}_0\rangle$.
- Step4 : Observe the sub-system $\mathbb{H}^{I/O} = ¶\mathbb{H}^{\otimes n}$ and read off the observed $x \in \{0, \cdots, N-1\}$.
- Step5: By evaluating $g(x)$, check if $x \in S$.
- Output : A solution $x \in S$ with the probability no less than $\frac{1}{4}$ (Theorem 6.32).

The End