量子光学学报 19(2)：129～133，2013
Acta Sinica Quantum Optica

文章编号：1007-6654(2013)02-0129-05

量子密钥分发私密放大的实现

杜鹏燕， 白增亮， 王旭阳， 李永民†

（量子光学与光量子器件国家重点实验室，山西大学光电研究所，山西 太原 030006）

摘要：在量子密钥分发系统中，私密放大是合法通信双方提取共享安全密钥的一个必不可少的环节。本文主要介绍了实现私密放大的两种加速算法，采用将两种加速算法进行有效组合的方法，既可以保证私密放大的安全性，同时又有效地减少了运算的操作次数。在连续变量量子密钥分发实验系统中，采用上述组合加速算法，实现了安全密钥的提取。

关键词：量子密钥分发； 私密放大； Hash 函数； 数论变换

中图分类号：O431 **文献标识码**：A **doi**：10.3788/ASQO20131902.0129

0 引言

量子密钥分发是通信双方 Alice 和 Bob 共享安全密钥的过程。在第三方窃听者 Eve 存在的情况下，为了保证量子密钥分发的安全性，Alice 和 Bob 必须从部分安全的比特串中提取出完全安全的密钥，这个过程就叫做私密放大[1]。

在图 1 所示的量子密钥分发系统中，合法通信双方之间经过数据协调之后，发送方 Alice 和接收方 Bob 之间共同拥有一组相同的比特串 W。由于在量子信道和经典信道中第三方窃听者 Eve 的存在，Eve 也会得到一组比特串 V，其中 V 包含有比特串 W 中的一部分信息。为了从共享的信息 W 中去除 Eve 所窃取的信息 V，从而得到一组完全安全的密钥 K，Alice 和 Bob 利用私密放大的方法，即通过与通用类 Hash 函数 $H:\{0,1\}^n \to \{0,1\}^k$ 相作用，使得 Eve 得到的信息量以指数级减少，从而极大地提高了密钥的安全性[2]。通用类的 *Hash* 函数是私密放大过程中的一个非常重要的工具，它首先由 Carter 和 Wegman 所提出[3]，最初主要用于数据的存储与恢复，后来又被用于身份认证[4]、数字签名等领域。1988 年，Bennett 等人将通用类的 Hash 函数用于私密放大过程中[1]。

本文结构如下，第一部分介绍了私密放大算法，重点介绍了两种加速算法及其实现过程，并分析了两种算法组合的具体步骤。第二部分分析了两种算法组合使用的安全性。第三部分给出了在我们的连续变量量子密钥分发实验系统中利用组合算法实现私密放大的过程和结果。第四部分是对本文的总结。

①收稿日期：2013-01-23

基金项目：国家自然科学基金(No.11074156)；山西省高等学校优秀青年学术带头人支持计划；山西省回国留学人员科研资助项目

作者简介：杜鹏燕(1986—)，女，山西长治人，硕士研究生，研究领域：量子通信。E-mail：dupengyan3214@163.com

†通讯作者：李永民，E-mail：yongmin@sxu.edu.cn

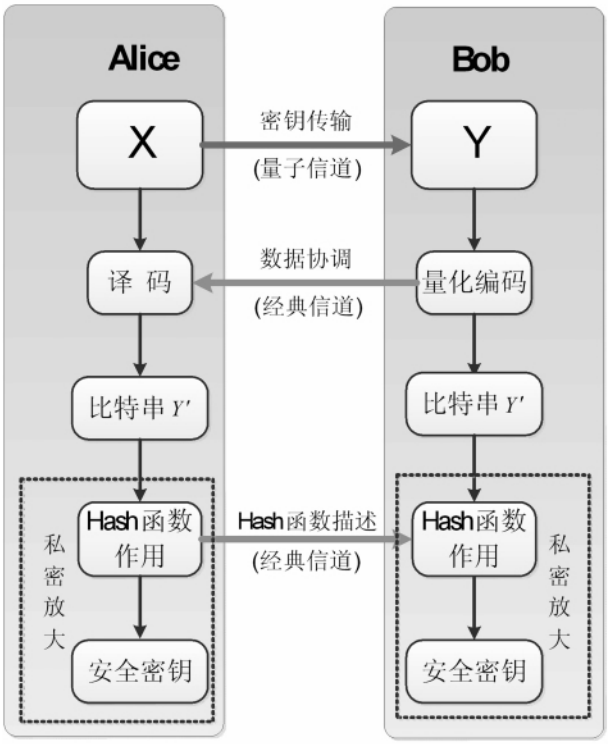

图 1 量子密钥分发系统

Fig. 1 The system of quantum key distribution

1 私密放大算法：有限域 $GF(2^l)$ 中乘法运算

通信双方 Alice 和 Bob 经过数据协调以后，得到一组完全相同的二进制串，然后通过以下几个步骤进行私密放大，提取安全的密钥。

(1)首先将数据协调以后得到的长度为 n 的比特串，通过对其进行补“0”操作，将其扩展成长度为 l 的比特串。然后再将其转化为多项式的表示形式：$P_{in}[X]$。

(2)Alice 和 Bob 随机地产生一个长度为 l 的比特串，作为 Hash 函数，并转化为多项式的表示形式：$P_h[X]$。

(3)Alice 和 Bob 分别在有限域 $GF(2^l)$ 中计算多项式的乘积：$P_{in}[X]\times P_h[X]$，并通过不可约多项式 x^l+x^s+1 化简，然后再转化为二进制比特串。其中不可约多项式 x^l+x^s+1 的度 l 对应于 Mersenne 素数 2^l-1 的指数 l。

(4)从上一步得到的比特串中提取出长度为 k 的安全密钥。

有限域 $GF(2^l)$ 中两个元素的乘法使用传统的移位相加运算法则，运算次数为长度 l 的平方次。所以随着长度 l 的增加，运算会消耗大量的时间。为了减少私密放大所消耗的时间，我们接下来介绍两种加速算法。

1.1 加速算法 1：采用快速数论变换(number-theoretic transform，NTT)加速

在这一部分中，主要介绍由 Gilles Van Assche 提出来的一种加速方法[5]，这种方法主要采用快速 NTT 对有限域 $GF(p)[X]/(X^L-1)$ 的多项式乘法运算进行加速。

首先定义一个有限域 $GF(p)[X]/(X^L-1)$ 中的多项式集合：

(1) p 是一个素数，L 是一个整数；

(2) 多项式的度必须小于 L；

(3) 乘法规则：多项式集合中的两个元素相乘以后，如果其结果的度不小于 L，则需对其进行模 X^L-1 运算，并且同时多项式的系数要进行模 p 运算。

在执行算法的过程中，我们不能任意地选择 L 和 p。其中应满足一定的条件：$L=2^m$(m 为整数)，$p=\nu L+1$(ν 为整数)，$p\geqslant l$ 以及 $L\geqslant 2l$。根据式子 $F(P_{in}\times P_h)=F(P_{in})\cdot F(P_h)$(其中 F 表示快速 NTT 操作)，我们首先对两个多项式进行快速 NTT 运算，然后再对其乘积做 NTT 的逆变换，最终得到乘积 $P_{in}\times P_h$。

1.2 加速算法 2：采用基于 NTT 的 Hash 函数类加速

前面所描述的加速算法，虽然非常有效，很大程度地加快了私密放大的运算时间，但不是最有效的方法。采用快速 NTT 算法，在运算过程中需要处理大约 $L\log_2 p$ bit。如果输入的比特串长度 $n=400\ 000$，则大约需要处理 50 Mbit 的数据。

下面介绍一种更有效的方法[5]。该方法是基于一种新的基于 NTT 的近通用类 Hash 函数。使用这类 Hash 函数进行私密放大，可以一次处

理的比特串长度为 $L\log_2 p$，而不仅仅是 $l(l<L/2)$。

定义：定义一个近通用类的 Hash 函数：$H_{p,L,\beta}=\{h_C:C_i\neq 0\ \forall\ i\}$。其中 $1\leqslant\beta\leqslant L,C,R\in Z_p^L$，且 $C_i\neq 0,\forall\ i=0,1,\cdots,L-1$。那么有 $h_C(R)=(F^{-1}(C\cdot R))_{0,1,\cdots,\beta-1}$。

定理：如果 $p-1\geqslant L$，$H_{p,L,\beta}$ 是一个近通用类 Hash 函数，其普适度 $\varepsilon=\left(\frac{p}{p-1}\right)^{\beta}$。

从以上的定义可知，Hash 函数的输入不再是二进制数，而是 L 个 0 到 $p-1$ 的整数。首先应将长度为 n 的比特串通过补“0”扩展成长度为 $L[\log_2 p]$ 的比特串（$L[\log_2 p]\geqslant n$），然后将比特串平均分成 L 份，每一份二进制串转化为一个取值为 0 到 $p-1$ 的整数。经过 Hash 函数作用以后，将每个整数转化为 $[\log_2 p]$ 个比特，直到提取出长度为 k 的安全密钥。

与前面的算法不用，算法 2 采用的是非通用类 Hash 函数，它的普适度为 $\varepsilon=\left(\frac{p}{p-1}\right)^{k}\approx 1+\frac{k}{p}$。因此，我们应合理地选择参数 p 和 L 的值。算法的运算时间会随着 L 的增大而变长，所以在保证条件 $L\log_2 p\geqslant n$ 满足的情况下，应该尽量选择比较小的 L 和比较大的 p。

1.3　加速算法 1 与 2 的组合

通过以上算法的分析介绍，我们知道算法 1 是通用类的 Hash 函数，但是运算次数比较多，算法 2 虽然是近通用类的 Hash 函数，其安全性方面稍有减弱，但是运算时间会大大减少。所以如果将上述两种算法合理组合起来，这样既可保证私密放大的安全性，同时又可以减少程序的运算时间。

首先，我们采用算法 2 将输入的长度为 n 的比特串经私密放大之后，得到长度为 i 的比特串。其中 i 应满足 $i>k+s$，s 为安全参量。然后，再采用算法 1 进行私密放大，从长度为 i 的比特串中提取出长度为 k 的安全密钥。

2　私密放大的安全性

在私密放大过程中，如果采用普适度为 ε 的近通用类的 Hash 函数作用，根据已知的定理[5]，有：

$$H(K\mid h,V)\geqslant k-\log_2\varepsilon-\frac{2^{-s-\log_2\varepsilon}}{\ln 2}\qquad(1)$$

其中 k 为最终提取的安全密钥长度，s 为安全参量。考虑到近通用类 Hash 函数非常接近于通用类 Hash 函数，也就是说 $\varepsilon-1\ll 1$，所以上式可以写成

$$H(K\mid h,V)\geqslant k-\frac{2^{-s}+\varepsilon-1}{\ln 2}\qquad(2)$$

根据公式 $I(K,hV)=H(K)-H(K\mid h,V)$ 可得：

$$I(K,hV)\leqslant\frac{2^{-s}+\varepsilon-1}{\ln 2}\qquad(3)$$

那么从上式可以看出，我们可以采用近通用类 Hash 函数作用，使得窃听者 Eve 获得的信息量尽可能地小，但安全参量要满足一个极限条件：

$$s<-\log_2(\varepsilon-1)\qquad(4)$$

在私密放大过程中，为了减小程序所消耗的时间，可以将两种加速方法组合起来[6]。首先使用 Hash 函数 H_a（普适度为 ε_a）将长度为 n 的比特串私密放大，得到长度为 i 的比特串，然后再使用 Hash 函数 H_b（普适度为 ε_b）将长度为 i 的比特串私密放大，得到长度为 k 的安全密钥。二者 Hash 函数类相结合的普适度写为 $\varepsilon=\varepsilon_a 2^{k-i}+\varepsilon_b$。在实验中我们选择通用类 Hash 函数 $H_b(\varepsilon_b=1)$ 以及近通用类 Hash 函数 $H_a(\varepsilon_a\approx 1)$，其中安全参量 s 可以取满足条件 $i>k+s$ 的任何值。

3　连续变量量子密钥分发私密放大的实现

在连续变量量子密钥分发过程中，Alice 和 Bob 经过量子传输、数据筛选、数据协调三个步骤之后，双方已经共享了一组完全相同的二进制比特串。根据实验参数可计算出有效的安全密钥速率 $\Delta I_{eff}=0.044$。我们以 $N=200\ 000$ 个信号

脉冲作为一次私密放大的处理单元，由于数据协调过程中连续变量采用 4－bit 量化[7]，其中有两级直接公开，所以私密放大每一次处理的比特串长度 $n=2N=400\ 000$。最大可提取的安全密钥长度 $k=N\Delta I_{eff}=8\ 800$。

私密放大的程序是基于 Matlab 编写，整个程序设计流程如图 2 所示。

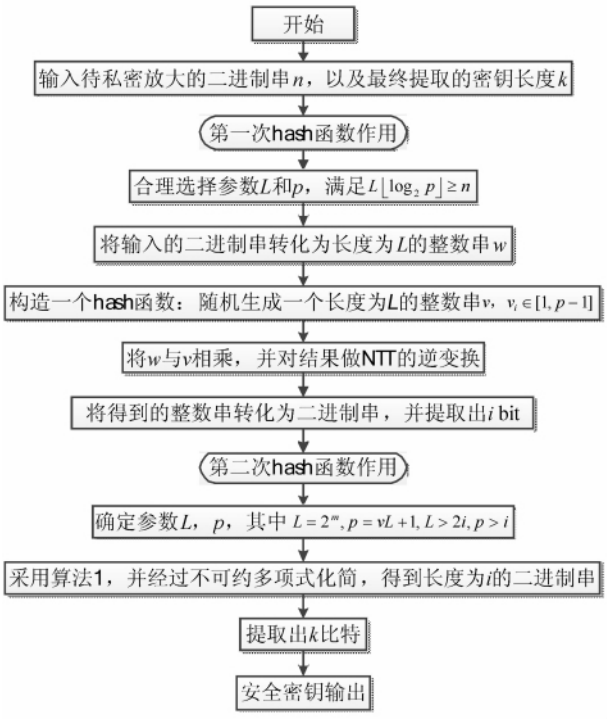

图 2 私密放大程序流程图

Fig. 2 Procedure of privacy amplification

在实验中，根据一次要处理的比特串长度 $n=400\ 000$，选定各个参数，其中 $m=14$，$L=16\ 384$，$v=2\ 065$，$p=33\ 832\ 961$，使得条件 $L\lfloor\log_2 p\rfloor=409\ 600>400\ 000$ 满足。首先采用算法 2 进行第一次私密放大，得到的比特串长度 $i=9\ 689$。然后我们采用算法 1 进行第二次私密放大，选定参数如下：$m=15$，$L=32\ 768$，$v=2$，$p=65\ 537$。最大可提取的安全密钥长度 $k=8\ 800$。根据以上安全性分析可知，两类 Hash 函数组合的普适度 $\varepsilon=\left(1+\frac{i}{p}\right)2^{k-i}+1\approx 2^{-889}+1$，安全参量 s 最大可以达到 889，此时安全密钥长度 $k=N\Delta I_{eff}-s=7\ 911$。

4 总结

本文主要介绍了量子密钥分发过程的私密放大方案，重点介绍了两种私密放大过程的加速方法，并具体分析了每一种加速算中各个参数值的选定。在私密放大过程中，将两种加速算法有效地组合起来，既保证了私密放大的安全性，又很大程度地降低了私密放大过程所消耗的时间。在连续变量量子密钥分发实验系统中，我们以 200 000 个信号脉冲作为一次私密放大的处理单元，采用两种加速算法的组合方法，最终能够提取出 8 800 bit 安全密钥。

参考文献：

[1] BENNETT C H, BRASSARD G, ROBERT J-M. Privacy Amplification by Public Discussion [J]. *SIAM J Comput*, 1988, **17**(2): 210-229.

[2] BENNETT C H, BRASSARD G, CREPEAU C, *et al*. Generalized Privacy Amplification [J]. *IEEE Trans Inform Theory*, 1995, **41**(6): 1915-1923.

[3] CARTER J L, WEGMAN M N. Universal Classes of Hash Functions [J]. *J Comput Syst Sci*, 1979, **18**: 143-154.

[4] WEGMAN M N, CARTER J L. New Hash Functions and Their Use in Anthentication and Set Equality [J]. *J Comput Syst Sci*, 1981, **22**: 265-279.

[5] ASSCHE G V. Quantum Cryptography and Secret-key Distillation [J]. *Cambridge University Press*, 2006, **89-112**.

[6] LODEWYCK J, BLOCH M, GARCÍA-PATRÓN R, *et al*. Quantum Key Distribution Over 25 km with an All-fiber Continuous-variable System [J]. *Phys Rev A*, 2007, **76**: 042305.

[7] 白增亮，王旭阳，杜鹏燕，等. 连续变量量子密钥分发的数据逆向协调 [J]. 量子光学学报，2012，**18**(1)：23-26.

Privacy Amplification for Quantum Key Distribution

DU Peng-yan, BAI Zeng-liang, WANG Xu-yang, LI Yong-min

(*State Key Laboratory of Quantum Optics and Quantum Optics Devices, Institute of Opto-Electronics, Shanxi University, Taiyuan* 030006, *China*)

Abstract: Privacy amplification is an essential part of distilling highly secret shared information in quantum key distribution. Two accelerating algorithms used for privacy amplification are introduced in this paper. The combination of the two accelerating algorithms not only has the merits of high security, but also effectively reduces the computation time. By using the combined algorithm, we have achieved secret-key distillation for continuous variable quantum key distribution.

Key words: quantum key distribution; privacy amplification; hash functions; number-theoretic transform