Optik 124 (2013) 2273–2276

Contents lists available at SciVerse ScienceDirect

Optik

journal homepage: www.elsevier.de/ijleo

Research of hash-based secure key expansion algorithm for practical QKD

Jian Wang[a,b,*], Chun-li Luo[a,b], Sheng-zhao Lin[a,b], Hong-fei Zhang[a,b], Ke Cui[a,b], Hao Liang[a,b], Ge Jin[a,b], Lei Zhou[b], Teng-yun Chen[b]

[a] Anhui Key Laboratory of Physical Electronics, Modern Physics Department, University of Science and Technology of China, State Key Laboratory of Particle Detection and Electronics, Hefei, Anhui 230026, China
[b] Hefei National Laboratory for Physical Sciences at Microscale, Hefei, Anhui 230026, China

ARTICLE INFO

Article history:
Received 14 February 2012
Accepted 25 June 2012

Keywords:
Quantum key distribution
Key expansion
Secure hash algorithm
Field programmable gate array

ABSTRACT

The quantum key distribution (QKD) system has been developed rapidly, but its key generation rate is limited for kinds of reason such as detector efficiency and not fitted for high speed application such as video conferences. For promotion of key generation rate, an algorithm based secure hash algorithm (SHA) is introduced to process QKD keys which could be expanded to be about tens times and implemented in field programmable gate array (FPGA) device in this paper. The expanded key is tested by NIST test program to verify its randomness and security. In our tests, the expanded keys less than 32 times QKD keys are all passed NIST test program and shows its good security.

1. Introduction

The system of quantum key distribution can ensure the obtainment of unconditional secure key [1,2] for its user in the resistance against any eavesdropping attack from the public channel, on the ground that its security is supported by essential principles of physics. In two decades, the practical QKD is rapidly developed and the high speed system also appeared with GHz transmitter repetition rate and Mbps key rate [3], but it is too expensive for normal applications. For the terminal application, the transmitter frequency of QKD system is from 4 MHz to 20 MHz, and the key rate is from 1 kbps to 10 kbps. If the key rate is less than 5 kbps, for the audio application, the system need an audio chip with high data-compressed algorithm and less quality of voice. If the system needs a high quality of voice for audio application, it should use a low compressed algorithm for voice data compressing and also need higher key rate for QKD system. In 2008, we implemented a quantum call with a QKD system that the transmitter frequency is 4 MHz and the key rate is about 1 kbps, the data speed of audio chip 600 bps [3]. In 2009, we established a QKD network with 3–4 kbps key rate and its audio application could use better audio compress chip with 1.2 kbps [4]. Now we are constructing a wide QKD network with 20 MHz transmitter and about 10 kbps key rate [5]. This system can be used for audio application and can get much better quality of voice. But if it wants to be used for a real time video application, it needs a higher key rate.

For a real time video application such as a video conference, it needs more than 100 kbps communication speed, such as the H.323 standard defined the transmission rate is more than 128 kbps and optimal rate is more than 384 kbps. So we need a key expansion algorithm for QKD system with 20 MHz transmitter, and design a secure key expansion algorithm base on secure hash algorithm which could be implemented quasi-OTP [6] application with high security.

In this paper, we introduce the key expansion algorithm and its implement in FPGA device, then security of algorithm is analyzed, at last we give the test result of our key expansion algorithm with different multiples such as 2 times, 4 times, 8 times, 16 times and 32 times which shows the expanded keys have good randomness and proof its high security.

2. SHA-based key expansion and implemented in FPGA

For secure application with high transmission rate while key rate of QKD system is low, a secure key expansion algorithm is design to expand the keys from QKD system. The expanded key is used in the secure communication application with One-Time-Pad cipher. If the key is safe, the communication is safe.

2.1. Key expansion algorithm

The hash function has strong anti-collision ability, its output cover uniform distribution and high sensitive for the input data [7]. The key expansion algorithm is designed based on hash function.

* Corresponding author at: Anhui Key Laboratory of Physical Electronics, Modern Physics Department, University of Science and Technology of China, Room 405, No. 96 Jin Zhai Road, Building of Modern Physics Department, Hefei, Anhui 230026, China. Tel.: +86 551 3607447.
E-mail address: wangjian@ustc.edu.cn (J. Wang).

0030-4026/$ – see front matter © 2012 Elsevier GmbH. All rights reserved.
http://dx.doi.org/10.1016/j.ijleo.2012.06.087

According the computing ability of modern computer, strong non-collision is called that the probability of collision of hash output is less than $1/2^{64}$. That means we have no less than 2^{64} times to calculate hash values, it could be one match called one collision. So the multiple of expanded key could be very high such as 2^{64} based on hash function, but for algorithm implement, it could not be so much and normally is tens or hundreds about. On the other hand, the key from QKD system is a subset of true random number set, and after key expansion, expanded key must have randomness for security.

Secure hash algorithm (SHA) [8] is a secure hash function, which has been proofed. The keys processed with SHA have good randomness and statistical characteristics. SHA is an algorithm family which was designed by National Security Agency (NSA) and published by National Institute of Standards and Technology (NIST). SHA-2 is consists of SHA-224, SHA-256, SHA-384 and SHA512 which have no efficient collision attacks. In this paper, SHA-256 is selected for demonstration and practical application.

The length of input data of SHA-256 algorithm is less than 2^{64} bits and the length of output is fixed as 256 bits. Algorithm of SHA-256 has steps as follows: 1. padding the message; 2. parsing the padded message; 3. setting the initial hash value; 4. hash computation; and 5. output final data digest.

SHA-256 uses six basic logical functions, where each function operates on 32-bit words, which are represented as x, y, and z. The result of each function is a new 32-bit word. These functions are used in step 4 to hash computation. The six basic functions are as follows:

$$\mathrm{Ch}(x, y, z) = (x \wedge y) \oplus (-x \wedge z) \tag{1}$$

$$\mathrm{Maj}(x, y, z) = (x \wedge y) \oplus (x \wedge z) \oplus (y \wedge z) \tag{2}$$

$$\sum_0^{\{256\}}(x) = \mathrm{ROTR}^2(x) \oplus \mathrm{ROTR}^{13}(x) \oplus \mathrm{ROTR}^{22}(x) \tag{3}$$

$$\sum_1^{\{256\}}(x) = \mathrm{ROTR}^6(x) \oplus \mathrm{ROTR}^{11}(x) \oplus \mathrm{ROTR}^{25}(x) \tag{4}$$

$$\sigma^{\{256\}}(x) = \mathrm{ROTR}^7(x) \oplus \mathrm{ROTR}^{18}(x) \oplus \mathrm{ROTR}^3(x) \tag{5}$$

$$\sigma^{\{256\}}(x) = \mathrm{ROTR}^{17}(x) \oplus \mathrm{ROTR}^{19}(x) \oplus \mathrm{ROTR}^{10}(x) \tag{6}$$

In the hash computation step, the outer repetition is only once since the length of input data is less than 256 bits. And the intermediate value of inner repetition will be used for key expansion. A 32-bit constant Kt and a 32-bit register Wt are used in inner repetition of SHA-256. Wt is output of function whose inputs are 512-bits parsed from padded message. A 256-bit register is used for caching result of every inner repetition and final output of SHA-256. After every repetition, the register will be updated.

The key expansion based on SHA-256 has two methods. The first one is that SHA-256 is used directly with different inputs such as 2-bytes, 4-bytes, 8-bytes, 16-bytes according to 16 times, 8 times, 4 times, 2 times with input QKD keys.

The method two is for getting more multiples and more expanded keys, a algorithm based on SHA-256 called modified SHA-256 is designed that key data with fixed length is input such as 64 bits (better than 64 bits and more strong non-collision) and output is consists of the intermediate value of inner repetition and final digest of SHA-256. For example, the input is 64 bits key and output is 512 bits which is consists of 256 bits of no. 64 repetition value of SHA-256 and 256 bits of final digest of SHA-256. So we could use every repetition value of SHA-256 which is 256 bits, so the max output is 16,384 bits theoretically which are 256 multiple 64. If the input is 64 bits, the max multiple is 256. For security reason, we select some small such as about tens in fact.

We implemented two methods based on SHA-256 in FPGA device, which has high performance and high-speed processing.

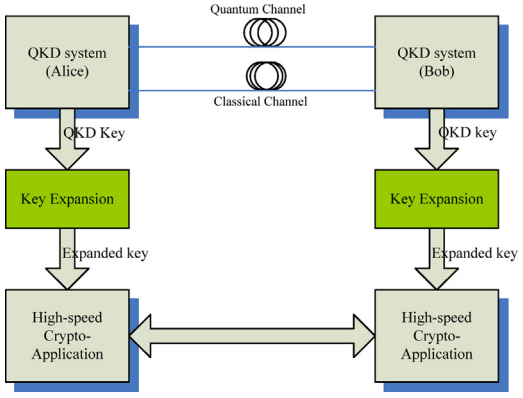

Fig. 1. Diagram of key expansion system.

The expanded keys are taken to process randomness test used NIST test program for security verification of expanded keys.

2.2. Implemented in FPGA device

The key expansion system is shown as Fig. 1. The whole system we established have implemented in FPGA device.

The main algorithm of SHA-256 is logical operation such as bit SHIFT, bit AND, bit XOR which could be programmed simply by hardware description language (HDL) such as Verilog and running very fast in FPGA device. So implement of key expansion algorithm is high performance and have enough output speed for high-speed crypto-application.

The structure of key expansion algorithm is shown in Fig. 2. The keys from QKD will be taken into 16-bit width FIFO, and processed by key expansion algorithm called modified SHA-256 described in Section 2.1 and the output is expanded key whose length is 256n bits, where n is a factor depends on the detail of expansion algorithm.

In the modified SHA-256 module, the input is 16n bits data from front FIFO that if input is 64 bits then $n = 4$, if input is 32 bits then $n = 2$, etc. Then initialize the data with SHA-256 padding message function. Then read the padded message, initialize Wt and Kt, next is 64 repetitions of operation which could be tapped as a part result of the expanded keys. After 64 repetitions, the SHA digest will be got through hash computation. The 256n bits expanded key then will be transmitted to high-speed crypto-application or to PC to test randomness of expanded keys.

In our design, which programmed with verilog HDL and running on a FPGA board as shown in Fig. 3. Only 66 clocks are used for one time key expansion and max work frequency is 80M on altera cyclone 3 series FPGA chip. The logic resources used in our design are only 6997 logic elements while total logic elements with all components such as QKD simulator, USB interface, and modified SHA-256 algorithm module are 8722.

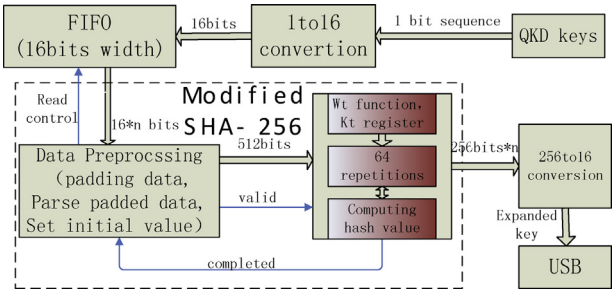

Fig. 2. Key expansion based on SHA-256 in FPGA device.