SCIENTIFIC REPORTS
nature research

OPEN

High-speed and Large-scale Privacy Amplification Scheme for Quantum Key Distribution

Bang-Ying Tang[1,4], Bo Liu[2,4], Yong-Ping Zhai[2], Chun-Qing Wu[3*] & Wan-Rong Yu[1*]

State-of-art quantum key distribution (QKD) systems are performed with several GHz pulse rates, meanwhile privacy amplification (PA) with large scale inputs has to be performed to generate the final secure keys with quantified security. In this paper, we propose a fast Fourier transform (FFT) enhanced high-speed and large-scale (HiLS) PA scheme on commercial CPU platform without increasing dedicated computational devices. The long input weak secure key is divided into many blocks and the random seed for constructing Toeplitz matrix is shuffled to multiple sub-sequences respectively, then PA procedures are parallel implemented for all sub-key blocks with correlated sub-sequences, afterwards, the outcomes are merged as the final secure key. When the input scale is 128 Mb, our proposed HiLS PA scheme reaches 71.16 Mbps, 54.08 Mbps and 39.15 Mbps with the compression ratio equals to 0.125, 0.25 and 0.375 respectively, resulting achievable secure key generation rates close to the asymptotic limit. HiLS PA scheme can be applied to 10 GHz QKD systems with even larger input scales and the evaluated throughput is around 32.49 Mbps with the compression ratio equals to 0.125 and the input scale of 1 Gb, which is ten times larger than the previous works for QKD systems. Furthermore, with the limited computational resources, the achieved throughput of HiLS PA scheme is 0.44 Mbps with the compression ratio equals to 0.125, when the input scale equals up to 128 Gb. In theory, the PA of the randomness extraction in quantum random number generation (QRNG) is same as the PA procedure in QKD, and our work can also be efficiently performed in high-speed QRNG.

Quantum Key Distribution (QKD), which based on the fundamental quantum mechanics, can generate the information-theoretical secure (ITS) keys for distant communication parties[1–3]. Practical QKD systems are mainly composed of two phases: the quantum communication phase and the post-processing phase[4,5]. In the post-processing phase, partial information about the secure key may still be leaked to the eavesdropper Eve after the key/basis sifting and error correction procedures. Privacy amplification (PA), the most significant post-processing procedure, coverts the weak secure correlated key to a uniform and ITS key to Eve[6–8].

Given the input weak secure key W with length of n and the security level ε, the optimal PA scheme in theory can be achieved with (dual) universal hash functions using Toeplitz kind of matrix (T) with computational complexity of $O(n\log n)$[9], and the length of consumed random seed in PA is αn, with min-entropy of $\alpha n + O(1)$, $\alpha \in (0,1]$[10–13].

Nowadays, state-of-art academic QKD experiments are performed with several GHz pulse rates[14–18], advanced multiplexing technologies[19,20] and extracts secure keys even with high-dimensional scenarios[21–23]. Meanwhile, a rigorous statistical fluctuation analysis has to be performed to remove the finite-size key effects on the final secure key[24,25]. Therefore, a high throughput and large-scale (usually larger than several Megabits) PA scheme has to be implemented to real-time extract the secure key with achievable generation rate close to the asymptotic (infinite-key) limit.

The simplest implementation idea of a large-scaled PA scheme is directly performing multiplication operation between W and T, resulting in the computational complexity with $O(n^2)$. However, such matrix-vector multiplication is very suitable to be implemented with Field-Programmable Gate Array (FPGA) platform. H. Zhang *et al.* firstly divided T into many smaller blocks and proposed a block parallel PA scheme to speedup the Toeplitz hashing procedure[26]. S. Yang *et al.*[27] and J. Constantin *et al.*[28] proposed advanced block partition strategies to

[1]College of Computer, National University of Defense Technology, Changsha, 410073, China. [2]College of Advanced Interdisciplinary Studies, National University of Defense Technology, Changsha, 410073, China. [3]Cyberspace Institute of Advanced Technology, Guangzhou University, Guangzhou, 510006, China. [4]These authors contributed equally: Bang-Ying Tang and Bo Liu. *email: 1405811286@qq.com; wlyu@nudt.edu.cn

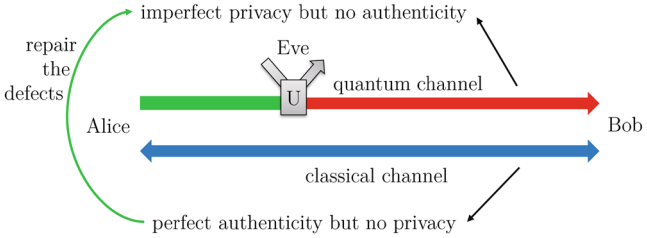

Figure 1. Schematic diagram of privacy amplification in quantum key distribution.

reduce the overhead of multiplication operations respectively, resulting in the throughput around 64 Mbps with input scale of 1 megabits[27].

Actually, majority optimized PA schemes are performed using fast Fourier transform (FFT) with complexity reduced to $O(n\log n)$[8,29,30]. Given fixed security level ε (i.e. 10^{-10}), the farther communication distance, the larger input length of PA scheme should be adapted. For example, in entanglement-based QKD systems, the input length n should be increased to at least the order of 10^8. B. Liu *et al.* firstly improved the throughput of FFT enhanced PA scheme to 60 Mbps with input scale of 12.8 megabits on Many-Integrated-Core (MIC) platform[8]. Z. L. Yuan *et al.* implemented a number theoretical transform (NTT) based PA scheme with throughput up to 108.77 Mbps with the input scale of 100 megabits also on MIC platform[31]. X. Wang *et al.* proposed a parallel implementation of the length-compatible (up to 10 Gbits) FFT based PA algorithm for continuous-variable QKD systems on a graphic processing unit (GPU) platform, with speed over 1 Gbps[29].

It's a huge challenge to implement large-scale FFT based PA schemes on FPGA platforms due to the limited resources and ultra complicated hardware design. Implementation of PA schemes on MIC, GPU or other dedicated computational devices consumes ultra high power and volume and significantly increases the design complexity. Improving the throughput of FFT enhanced PA schemes on CPU platforms is a very conventional option, since it can be efficiently integrated to the whole QKD system. However, it's feasible with CPU implementations only for small input scales ($\leq 10^6$) and rapidly becomes the performance bottleneck with larger input scales. Therefore, in this article, we propose a fast Fourier transform (FFT) enhanced high-speed and large-scale (HiLS) PA scheme on commercial multi-core CPU platform. In the HiLS PA scheme, W is divided into many blocks and the random seed for constructing Toeplitz matrix T is shuffled to multiple sub-sequences respectively, then PA procedures are parallel implemented for all sub-key blocks with correlated sub-sequences, afterwards the outcomes are merged as the final secure key. When the input scale is 128 Mb, our HiLS PA scheme reaches 71.16 Mbps, 54.08 Mbps and 39.15 Mbps with the compression ratio equals to 0.125, 0.25 and 0.375 respectively. Therefore, HiLS PA scheme can be applied to 10 GHz QKD systems with even larger input scales and the evaluated throughput is around 32.49 Mbps with the compression ratio equals to 0.125 and the input scale of 1 Gb, which is ten times larger than the previous works for QKD systems. Furthermore, with the limited computational resources (128 GB memory, 1 TB storage and 16 CPU cores in total), the achieved throughput of HiLS PA scheme is 0.44 Mbps with the compression ratio equals to 0.125, when the input scale equals up to 128 Gb. In theory, the PA of the randomness extraction in quantum random number generation (QRNG) is same as the PA procedure in QKD[32–34]. Thus, HiLS PA scheme can also be efficiently performed in high-speed QRNG.

Related Work

Privacy amplification was first proposed in the context of quantum key distribution by Bennett *et al.*[6], where the channel with perfect authenticity but no privacy (public classical channel) can be used to repair the defects of a channel with imperfect privacy but no authenticity (quantum channel). The schematic diagram of PA in QKD is shown in Fig. 1, Alice and Bob firstly distribute quantum signals via a noisy and lossy quantum channel (fiber or free space), then share correlated and weak secure key W after basis/key sifting and error correction procedures via a public channel. The min-entropy of shared weak secure key W is n. Let random variable E summarizes Eve's entire learned knowledge about W, here, $H(W|E) \leq t$, $t < n$. PA, where Alice and Bob publicly discuss a extractor function $G:\{0,1\}^n \rightarrow \{0,1\}^r$, such that reduces Eve's learned information of the final secure key K_f from t to at most ε[6,7,35,36]. Nowadays, most practical extractors are known to the universal hash function, especially the (modified) Toeplitz matrix defined as[13]

$$G(A) := (I_r|T(A)) = \begin{bmatrix} 1 & & & a_{r-1} & a_r & \cdots & a_{n-2} \\ & 1 & & a_{r-2} & a_{r-1} & \cdots & a_{n-3} \\ & & \ddots & \vdots & \vdots & \ddots & \vdots \\ & & & 1 & a_0 & a_1 & \cdots & a_{n-r-1} \end{bmatrix}, \quad (1)$$

where $T(A)$ is a $r \times (n-r)$ Toeplitz matrix, A is a random seed, $A = (a_0, a_1, \ldots, a_{n-1}) \in \{0,1\}^{n-1}$, $T(A)_{ij} = a_{j-i+r-1}$. Also, we define $W_I = (w_0, w_1, \ldots, w_{r-1})$ and $W_{TA} = (w_r, w_{r+1}, \ldots, w_{n-1})$. Therefore, the final secure key can be calculated as

$$K_f = G(A)W = I_r \times (w_0, w_1, \ldots, w_{r-1}) \oplus T(A) \times (w_r, w_{r+1}, \ldots, w_{n-1}) = W_I \oplus T(A)W_{TA}. \quad (2)$$

Parameter	Values
Pulse Repetition Rate ν_s	10 GHz
Heralding Efficiency	0.316
Dark Count Rate p_d	10^{-7}
Detector Efficiency η_d	0.40
Misalignment Error Rate e_d	0.015
Error Correction Efficiency f	1.10
Photon Pair Number per Coincidence Window μ	Optimal
Basis Reconciliation Factor q	0.50
Phase Error Estimation Failure Probability ε^{ph}	10^{-10}

Table 1. Parameters used for simulation of entanglement based QKD.

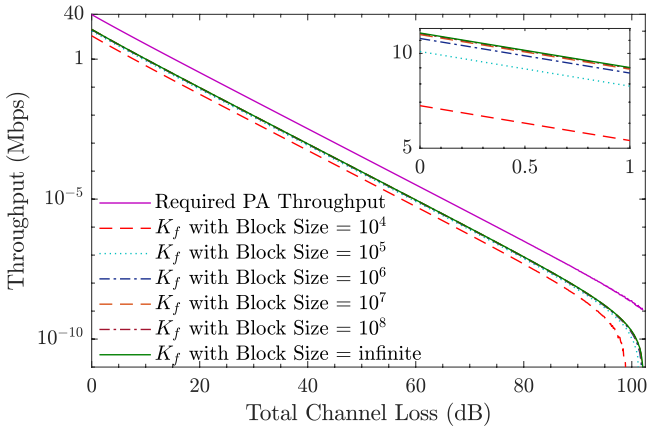

Figure 2. Required throughput of PA algorithms and final secure key rate with different block sizes for 10 GHz entanglement based QKD systems, under the simulation parameters shown in Table 1.

In order to efficiently implement the calculation of $T(A)W_{\mathrm{TA}}$ using fast Fourier transform (FFT), we have to extend $T(A)$ to a special circulant Toeplitz matrix with scale of $(n-1)\times(n-1)$ and extend W_{TA} to a vector with length of $n-1$ by padding zeros. The optimized multiplication of a circulant matrix and a vector is shown as

$$H \cdot X = F^{-1}[F(h) \ * \ F(X)], \tag{3}$$

where "*" denotes the Hadamard product operator, F denotes the Fourier transform operator, F^{-1} is the inverse Fourier transform operator, X is a vector and H is a circulant Toeplitz matrix with first row h. Since the complexity of F and F^{-1} operations is $O(n\log n)$ and the complexity of Hadmard product operation is $O(n)$, the computational complexity of optimized PA algorithm is $O(n\log n)$[8,12].

In theory, QKD can generate ITS keys for communication parties, even the quantum channel is under control of the eavesdropper Eve. Imperfect implementation and active attacks would leak some information about W to Eve. Alice and Bob can quantify the bound of leaked information accurately with the infinite post-processing block size. In this paper, we take entanglement based QKD as an example, the secure key rate can be calculated as[37]

$$R \geq qQ_\mu\nu_s[1 - H_2(e_p^U) - f(e_b)H_2(e_b)], \tag{4}$$

where q is the basis sifting factor, Q_μ is the gain of detected entangled photon pairs, ν_s is the repetition rate of the entangled source, e_b is the measured quantum bit error rate, e_p^U is the estimated upper-bound of phase error rate, $f(x)$ is the error correction efficiency, $H_2(x)$ is the binary Shannon entropy.

In practice, $e_p{}^U$ can not be measured directly and could not be accurately estimated due to the statistical fluctuations with finite post-processing block sizes. Here, we simulate the required throughput of PA algorithm in a 10 GHz entanglement based QKD with the parameters shown in Table 1. The entangled photon source is put into the middle of communication parties, the finite-size-effect for the final secure key K_f is considered with post-processing block size from the order of 10^4 to infinite, and the failure probability $\varepsilon^{ph} = 10^{-10}$ for estimating e_p^U[4]. The analyzed results are shown in Fig. 2, the post-processing block size should be at least the order of 10^8 to achieve a secure key rate close to the asymptotic limit. Directly implementing PA algorithms with ultra large-scale inputs will limit the performance of full QKD systems. Meanwhile, the required throughput of PA algorithm is around 40 Mbps without any channel loss.

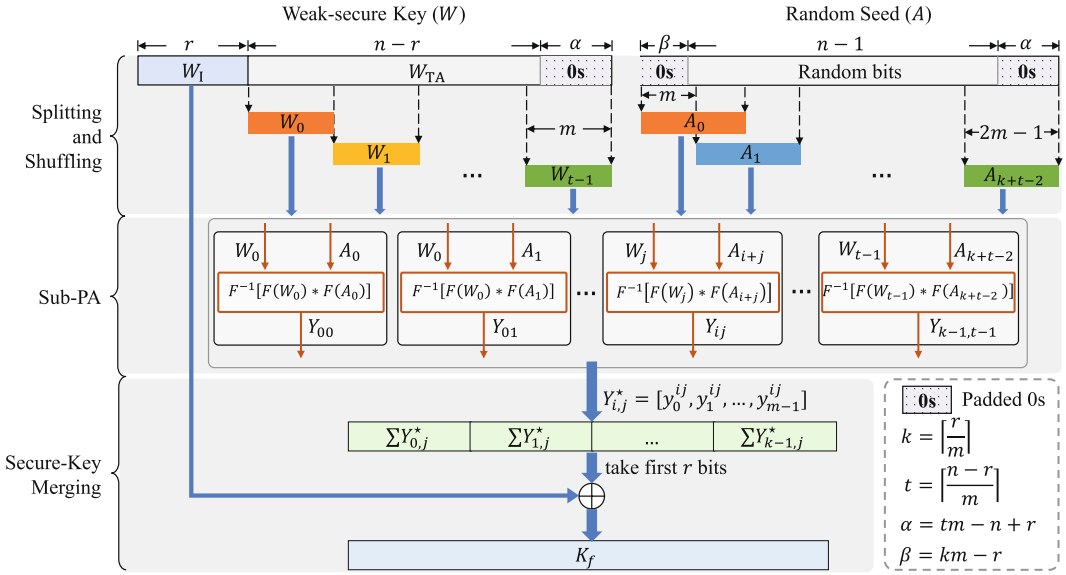

Figure 3. Schematic diagram of proposed high-speed and large-scale privacy amplification scheme for QKD. The weak secure key length is n, the final secure key length is r, the sub-block size is m, $0 < m \leq r < n$, $t = \lceil \frac{n-r}{m} \rceil$, $k = \lceil \frac{r}{m} \rceil$. $Y_{i,j} = F^{-1}[F(A_{i+j}) * F(W_j)]$, where "*" denotes the Hadamard product operator, F denotes the Fourier transform operator, F^{-1} is the inverse Fourier transform operator. $Y^{a}_{i,j}$ is a sub-vector consisted by first m bits of $Y_{i,j}$, defined as $Y^{a}_{i,j} = [y^{ij}_0, y^{ij}_1, \ldots, y^{ij}_{m-1}]$.

High-speed and Large-scale Privacy Amplification Scheme

The schematic diagram of proposed high-speed and large-scale (HiLS) privacy amplification scheme for QKD is shown in Fig. 3. Weak secure key W with length of n is gained after the basis/key sifting and error correction procedures for the measured raw key string at Alice's (Bob's) side. Then, Alice and Bob estimate the final secure key length r with rigorous statistical fluctuation analysis procedure. Afterwards, Alice and Bob publicly discuss a random seed with length of $n-1$ bits to construct the universal hash function. Our proposed HiLS PA scheme mainly consists of three steps: splitting and shuffling, sub-PA and secure-key merging.

Step 1: Splitting and shuffling. In this step, we divide W to several sub-vectors and divide the Toeplitz matrix $T(A)$ to sub-matrices. Assume the scale of sub-matrix is $m \times m$, $m \leq r$. Assume that the Toeplitz matrix $T(A)$ can be divided into t blocks by rows and k blocks by columns, thus in total kt sub-matrices, $t = \lceil \frac{n-r}{m} \rceil$, $k = \lceil \frac{r}{m} \rceil$. First of all, we construct a vector A by padding $km - r$ ($tm - n + r$) zeros to the head (tail) of the exchanged random seed with length of $n-1$ bits. Then, we shuffle A into $k + t - 1$ sub-vectors, defined as $A_i := [a_{im}, a_{im+1}, \ldots, a_{(2+i)m-1}]$, $0 \leq i < k + t - 1$. Therefore, the divided sub-matrix can be constructed by $H_{i,j} = T(A_{i+j})$, $i \in [0, k)$ and $j \in [0, t)$, and we have

$$T(A) = \begin{bmatrix} H_{k-1,0} & H_{k-1,1} & \cdots & H_{k-1,t-1} \\ H_{k-2,0} & H_{k-2,1} & \cdots & H_{k-2,t-1} \\ \vdots & \vdots & \ddots & \vdots \\ H_{00} & H_{01} & \cdots & H_{0,t-1} \end{bmatrix} = \begin{bmatrix} T(A_{k-1}) & T(A_k) & \cdots & T(A_{k+t-2}) \\ T(A_{k-2}) & T(A_{k-1}) & \cdots & T(A_{k+t-3}) \\ \vdots & \vdots & \ddots & \vdots \\ T(A_0) & T(A_1) & \cdots & T(A_{t-1}) \end{bmatrix}, \tag{5}$$

where $H_{i,j} = H_{i+1,j+1}$.

For W, we first pad $tm - n + r$ zeros to the tail and take first r bits and the rest bits to construct the sub-vector W_I and W_{TA}. Then, divide W_{TA} into t sub-vectors, defined as $W_i := [w_{im+r}, w_{im+r+1}, \ldots, w_{(i+1)m+r-1}]$, where $0 \leq i < t$.

Step 2: Sub-PA. In this step, the efficient implementation using FFT of multiplication $Y_{i,j}$ is performed to sub-vector W_j and sub-matrix $H_{i,j}$,

$$Y_{i,j} := F^{-1}\left[F(A_{i+j}) \ * \ F(W_j)\right], \tag{6}$$

where, $i \in [0, k)$ and $j \in [0, t)$.

Step 3: Secure-Key merging. First, we only take first m bits of $Y_{i,j}$ (defined as $Y^{a}_{i,j}$), then we merge $Y^{a}_{i,j}$ to vector Z by

$$Z = \left(\sum^{t-1} y^{*}_{0,j} | \sum^{t-1} y^{*}_{1,j} | \ldots | \sum^{t-1} Y^{*}_{k-1,j}\right). \tag{7}$$

Take first r bits of Z (defined as Z^{*}), we can get the final secure key K_f by

Parameter	Values
Operation System	CentOS 7
CPU	Intel(R) E5-2640 v3 × 2
Cores per CPU	8
Threads per core	2
Memory	128 GB
Storage	1 TB
Compiler	gcc 4.8.5
MPI	openmpi 1.10.7
FFT library	fftw 3.3.8

Table 2. Specifications of server computer.

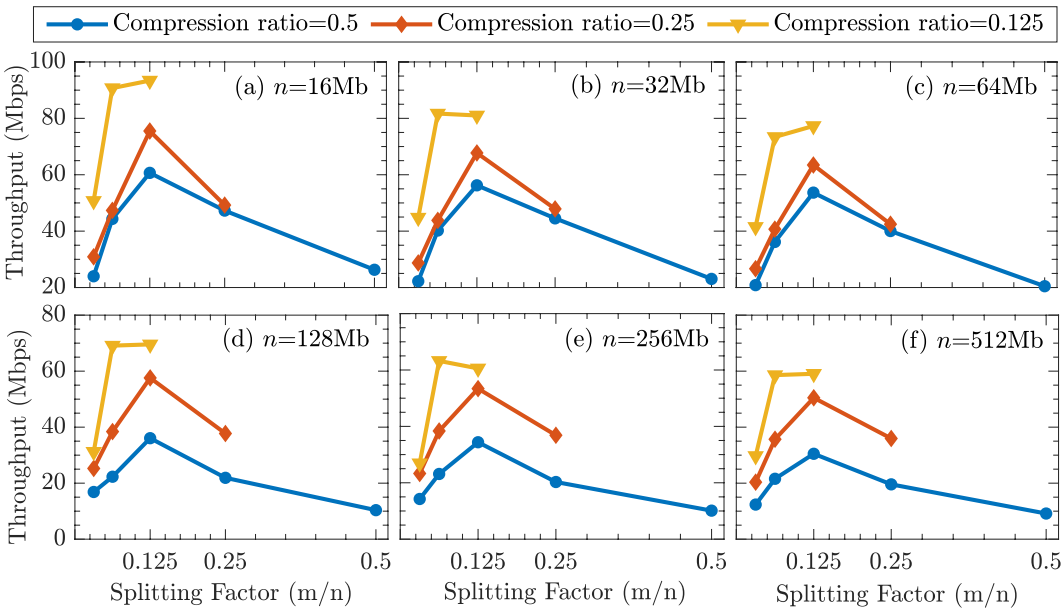

Figure 4. Throughput of HiLS PA scheme with n equals from 16 Mb to 512 Mb, and the splitting factor $\frac{m}{n}$ varies from $\frac{1}{32}$ to $\frac{1}{2}$.

$$K_f = W_1 \oplus Z^*. \tag{8}$$

The detailed implementation of HiLS PA scheme can be described as Algorithm 1. In the procedure of our proposed HiLS PA scheme, we only need to perform $k+2t-1$ times Fourier operations with scale of $2m$, kt times Hadamard product operations with scale of m, kt times inverse Fourier operations and $kt+1$ times exclusive or (XOR) operations with scale of m. Thus, the computational complexity of the proposed HiLS PA scheme is $O(ktm\log m)$, simplified to around $O(n\log m)$.

Algorithm 1. HiLS Privacy Amplification algorithm.

Input: the weak secure key W with length of n, the final key length r, the random seed S with length of $n-1$ and the sub-block size m

Output: the final secure key K_f

$t = \lceil \frac{n-r}{m} \rceil$, $k = \lceil \frac{r}{m} \rceil$

$W = (W|(\mathbf{0})_{tm-n+r})$, $A = ((\mathbf{0})_{km-r}|S|(\mathbf{0})_{tm-n+r})$

$W_1 = (w_0, w_1, \ldots, w_{r-1})$, $W_{\mathrm{TA}} = (w_r, w_{r+1}, \ldots, w_{tm-1})$

$\forall i \in [0,t)$, do $W_i = [w_{im+t}, w_{im+t+1}, \ldots, w_{(i+1)m+t-1}]$ and $P_i = F(W_i)$

$\forall i \in [0,k+t)$, do $A_i = [a_{im}, a_{im+1}, \ldots, a_{(2+i)m-1}]$ and $Q_i = F(A_i)$

while $0 \le i < k$ and $0 \le j < t$ **do**

  $Y_{i,j} = F^{-1}[Q_{i+j} * P_j]$

end while

$\forall i \in [0,k)$, do $Y^{\star}_{i,j} = \left[y_0^{ij}, y_1^{ij}, \ldots, y_{m-1}^{ij}\right]$ and $U_i = \sum_{j=0}^{t-1} Y^{\star}_{i,j}$

$Z = (U_0|U_1|\ldots|U_{k-1})$, $Z^{\star} = [z_0, z_1, \ldots, z_{r-1}]$

$K_f = W_1 \oplus Z^{\star}$

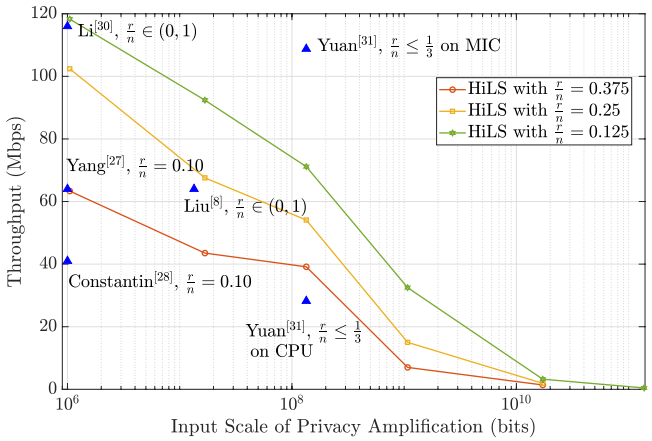

Figure 5. Comparison of HiLS PA scheme and privous works.

Results

The implementation of HiLS PA scheme is evaluated on the multi-core server computer, the specifications are shown in Table 2. Due to FFT operation may suffer errors caused by finite-precision float-point arithmetic, we suggest the scale of FFT operation smaller than the order of 10^8. Meanwhile, considering the thread synchronization and thread safety issues, the calculations of (inverse) Fourier transforms and also Hadamard products are paralleled in the architecture of shared memory multi-processes.

We evaluate the throughput of HiLS PA scheme with different input scale (n) and various sub-block size (m). The result is shown in Fig. 4, where we set the input weak secure key length n equals from 16 Mb to 512 Mb, and splitting factor, defined as $\frac{m}{n}$ is various from $\frac{1}{32}$ to $\frac{1}{2}$. Figure 4 shows us that for given n (in our implementation, can be up to 1 Gb), HiLS PA scheme can always achieve optimized throughput when splitting factor $\frac{m}{n} = 0.125$. When the splitting factor $\frac{m}{n} \leq 0.0625$, the Toeplitz matrix at least has to be divided into 28 sub-matrices with compression ratio $\frac{r}{n} \geq 0.125$, larger than 16 (amount of total cores), resulting HiLS PA scheme with very poor throughput due to heavy overhead of complicated process scheduling. When the splitting factor $\frac{m}{n} \geq 0.25$, less split sub-matrices ($\leq$4) only contributes a bit speedup to HiLS PA scheme, due to not fully used computational resource and still large scaled FFT operations. When the splitting factor $0.125 < \frac{m}{n} < 0.25$, the amount of split sub-matrices stays the level as the case with splitting factor equals to 0.125, but the FFT operating scale is same as the case with splitting factor equals to 0.25, which results even worse throughput to HiLS PA scheme. This situation would also happened when the splitting factor $0.0625 < \frac{m}{n} < 0.125$. For example, when $n = 512$ Mb, the optimized throughput of HiLS PA scheme is 59.06 Mbps, 50.48 Mbps and 30.49 Mbps when the compression ratio equals to 0.125, 0.25 and 0.50 respectively.

According to the simulation results shown in Fig. 2, the maximum compression ratio required for PA schemes is ($\frac{r}{n}$) is 0.297 for 10 GHz entanglement based QKD systems. Then, we optimized the implementation of HiLS PA scheme with $n = 1$ Mb, 16 Mb, 128 Mb and 1 Gb with compression ratio equals to 0.125, 0.25 and 0.375 respectively and compared with other previous works designed for QKD systems, e.g. entanglement based systems, the results are shown in Fig. 5. S. Yang *et al.*[27] and J. Constantin *et al.*[28] both implemented PA schemes on FGPA platform by performing multiplication operations, achieved 64.0 Mbps and 41.0 Mbps throughput with compression ratio equals to 0.10. Q. Li *et al.* achieved the throughput of 116.0 Mbps with adaptive compression ratio by implementing FFT operation on FPGA platform[30]. However, FPGA platform is not suitable for the implementation of PA schemes with ultra-large input scales (larger than the order of 10^8). B. Liu *et al.* achieved the throughput of 60 Mbps with input scale of 12.8 megabits by implementing the FFT enhanced PA scheme on MIC platform[8]. Z. L. Yuan *et al.* achieved the throughput of 108.77 Mbps with the input scale supported up to 128 megabits by implementing the NTT based PA scheme on MIC platform[31]. Z. L. Yuan *et al.* also evaluated the performance of their PA scheme on CPU platform, resulting in the throughput of 28.22 Mbps. When the input scale is 128 Mb, the finite-size-effect for the final secure key can be almost perfectly avoided, and the throughput of our proposed HiLS PA scheme reaches up to 71.16 Mbps, 54.08 Mbps and 39.15 Mbps with the compression ratio equals to 0.125, 0.25 and 0.375 respectively. In the case of input scale is 1 Gb, the throughput of HiLS PA scheme reaches up to 32.49 Mbps and 15.0 Mbps with the compression ratio equals to 0.125 and 0.25, which contributes much rigorous statistical fluctuation analysis and is remarkable higher than the required throughput when the total channel loss is expected larger than 87.6 dB.

With limited resource (128 GB memory, 1 TB storage and 16 CPU cores in total), the HiLS PA scheme with input scale of 128 Gb and the compression ratio equals to 0.125, runs around 83 hours, resulting a throughput of 0.44 Mbps. The implementation of PA with such large inputs on GPU platform is very difficult due to the complicated computation and memory scheduling strategies. Meanwhile, the throughput of the HiLS PA scheme can be easily improved on high-speed multi-core CPU platforms with much larger configured memory.

Conclusion

In this paper, we propose a fast Fourier transform (FFT) enhanced high-speed and large-scale (HiLS) PA scheme on multi-core CPU platform. The long input weak secure key is divided into many blocks and the random seed for constructing Toeplitz matrix is shuffled to multiple sub-sequences respectively, then PA procedures are parallel implemented for all sub-key blocks with correlated sub-sequences, afterwards the outcomes are merged as the final secure key. When the input scale is 128 Mb, our proposed HiLS PA scheme reaches 71.16 Mbps, 54.08 Mbps and 39.15 Mbps with the compression ratio equals to 0.125, 0.25 and 0.375 respectively, resulting achievable secure key generation rates close to the asymptotic limit. HiLS PA scheme can be efficiently implemented on the commercial CPU platform without increasing dedicated computational devices and can be applied to 10 GHz QKD systems with even larger input scales. The evaluated throughput of HiLS PA scheme is around 32.49 Mbps with the compression ratio equals to 0.125 and the input scale of 1 Gb, which is ten times larger than the previous works for QKD systems.Furthermore, with the limited computational resources, the achieved throughput of HiLS PA scheme is 0.44 Mbps with the compression ratio equals to 0.125, when the input scale equals up to 128 Gb. As randomness extraction with Toeplitz hashing in QRNG is particularly efficient, the HiLS PA scheme can be also performed in high-speed QRNG.

Received: 24 July 2019; Accepted: 3 September 2019;
Published online: 31 October 2019

References

1. Gisin, N., Ribordy, G., Tittel, W. & Zbinden, H. Quantum cryptography. *Reviews of modern physics* **74**, 145 (2002).
2. Scarani, V. *et al*. The security of practical quantum key distribution. *Reviews of Modern Physics* **81**, 1301–1350 (2009).
3. Lo, H.-K., Curty, M. & Tamaki, K. Secure quantum key distribution. *Nature Photonics* **8**, 595, https://doi.org/10.1038/nphoton.2014.149 (2014).
4. Fung, C.-H. F., Ma, X. & Chau, H. F. Practical issues in quantum-key-distribution postprocessing. *Physical Review A* **81**, 012318 (2010).
5. Ma, X., Fung, C.-H. F., Boileau, J.-C. & Chau, H. Universally composable and customizable post-processing for practical quantum key distribution. *Computers & Security* **30**, 172–177 (2011).
6. Bennett, C. H., Brassard, G. & Robert, J.-M. Privacy amplification by public discussion. *SIAM journal on Computing* **17**, 210–229 (1988).
7. Bennett, C. H., Brassard, G., Crépeau, C. & Maurer, U. M. Generalized privacy amplification. *IEEE Transactions on Information Theory* **41**, 1915–1923 (1995).
8. Liu, B., Zhao, B.-K., Yu, W.-R. & Wu, C.-Q. Fit-pa: Fixed scale fft based privacy amplification algorithm for quantum key distribution. *Journal of Internet Technology* **17**, 309–320 (2016).
9. Golub, G. H. & Loan, C. F. V. *Matrix computations*, third edition edn. (The Johns Hopkins University Press, 1996).
10. Carter, J. L. & Wegman, M. N. Universal classes of hash functions. *Journal of computer and system sciences* **18**, 143–154 (1979).
11. Mansour, Y., Nisan, N. & Tiwari, P. The computational complexity of universal hashing. *Theoretical Computer Science* **107**, 121–133, https://doi.org/10.1016/0304-3975(93)90257-T (1993).
12. Hayashi, M. Exponential decreasing rate of leaked information in universal random privacy amplification. *IEEE Transactions on Information Theory* **57**, 3989–4001 (2011).
13. Hayashi, M. & Tsurumaru, T. More efficient privacy amplification with less random seeds via dual universal hash function. *IEEE Transactions on Information Theory* **62**, 2213–2232, https://doi.org/10.1109/TIT.2016.2526018 (2016).
14. Gordon, K. J. *et al*. Quantum key distribution system clocked at 2 ghz. *Optics Express* **13**, 3015–3020, https://doi.org/10.1364/OPEX.13.003015 (2005).
15. Takesue, H., Diamanti, E., Langrock, C., Fejer, M. M. & Yamamoto, Y. 10-ghz clock differential phase shift quantum key distribution experiment. *Optics Express* **14**, 9522–9530, https://doi.org/10.1364/OE.14.009522 (2006).
16. Bienfang, J. C. *et al*. Quantum key distribution with 1.25 gbps clock synchronization. *Optics Express* **12**, 2011–2016, https://doi.org/10.1364/OPEX.12.002011 (2004).
17. Patel, K. A. *et al*. Quantum key distribution for 10 gb/s dense wavelength division multiplexing networks. *Applied Physics Letters* **104**, 051123 (2014).
18. Wang, S. *et al*. 2 ghz clock quantum key distribution over 260 km of standard telecom fiber. *Optics Letters* **37**, 1008–1010, https://doi.org/10.1364/OL.37.001008 (2012).
19. Fróhlich, B. *et al*. A quantum access network. *Nature* **501**, 69, https://doi.org/10.1038/nature12493 (2013).
20. Liu, H., Wang, J., Ma, H. & Sun, S. Polarization-multiplexing-based measurement-device-independent quantum key distribution without phase reference calibration. *Optica* **5**, 902–909, https://doi.org/10.1364/OPTICA.5.000902 (2018).
21. Mower, J. *et al*. High-dimensional quantum key distribution using dispersive optics. *Phys. Rev. A* **87**, 062322, https://doi.org/10.1103/PhysRevA.87.062322 (2013).
22. Canas, G. *et al*. High-dimensional decoy-state quantum key distribution over multicore telecommunication fibers. *Physical Review A* **96**, 022317, https://doi.org/10.1103/PhysRevA.96.022317 (2017).
23. Steinlechner, F. *et al*. Distribution of high-dimensional entanglement via an intra-city free-space link. *Nature Communications* **8**, 15971, https://doi.org/10.1038/ncomms15971 (2017).
24. Zhang, Z., Zhao, Q., Razavi, M. & Ma, X. Improved key-rate bounds for practical decoy-state quantum-key-distribution systems. *Physical Review A* **95**, 012333 (2017).
25. Cai, R. Y. & Scarani, V. Finite-key analysis for practical implementations of quantum key distribution. *New Journal of Physics* **11**, 045024 (2009).
26. Zhang, H.-F. *et al*. A real-time qkd system based on fpga. *Journal of Lightwave Technology* **30**, 3226–3234 (2012).
27. Yang, S. S. *et al*. Fpga-based implementation of size-adaptive privacy amplification in quantum key distribution. *IEEE Photonics Journal* **9**, 1–8 (2017).
28. Constantin, J. *et al*. An fpga-based 4 mbps secret key distillation engine for quantum key distribution systems. *Journal of Signal Processing Systems* **86**, 1–15, https://doi.org/10.1007/s11265-015-1086-1 (2017).
29. Wang, X., Zhang, Y., Yu, S. & Guo, H. High-speed implementation of length-compatible privacy amplification in continuous-variable quantum key distribution. *IEEE Photonics Journal* **10**, 1–9, https://doi.org/10.1109/JPHOT.2018.2824316 (2018).
30. Li, Q. *et al*. High-speed and adaptive fpga-based privacy amplification in quantum key distribution. *IEEE Access* **7**, 21482–21490, https://doi.org/10.1109/ACCESS.2019.2896259 (2019).
31. Yuan, Z. *et al*. 10-mb/s quantum key distribution. *Journal of Lightwave Technology* **36**, 3427–3433 (2018).
32. Ma, X., Yuan, X., Cao, Z., Qi, B. & Zhang, Z. Quantum random number generation. **2**, 16021, https://doi.org/10.1038/npjqi.2016.21 (2016).

33. Ma, X. *et al*. Postprocessing for quantum random-number generators: Entropy evaluation and randomness extraction. *Physical Review A* **87**, https://doi.org/10.1103/PhysRevA.87.062327 (2013).
34. Herrero-Collantes, M. & Garcia-Escartin, J. C. Quantum random number generators. *Reviews of Modern Physics* **89**, 015004 (2017).
35. Krawczyk, H. Lfsr-based hashing and authentication. In *Annual International Cryptology Conference*, 129–139 (Springer, 1994).
36. Asai, T. & Tsurumaru, T. Efficient privacy amplification algorithms for quantum key distribution. *IEICE Tech. Rep.* **110**, 327–332 (2011).
37. Ma, X., Fung, C.-H. & Lo, H.-K. Quantum key distribution with entangled photon sources. *Physical Review A* **76**, 012307 (2007).

Acknowledgements
This work was supported in part by the National High Technology Research and Development Program of China under Grant No. 2015AA1138 and the National Natural Science Foundation of China under Grant No. 61972410.

Author contributions
B.Y.T. and B.L. proposed the scheme, performed the experiments, wrote the paper and contributed equally. Y.P.Z. helped with the experimental implementation and results analysis. This work was conceived by B.L. and W.R.Y., supervised by W.R.Y. and co-supervised by C.Q.W. All authors reviewed the manuscript.

Competing interests
The authors declare no competing interests.

Additional information
Correspondence and requests for materials should be addressed to C.-Q.W. or W.-R.Y.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.