A Reconciliation Strategy for Real-Time Satellite-Based QKD

Xiaoyu Ai, Robert Malaney, and Soon Xin Ng

***Abstract*— In practical Quantum Key Distribution (QKD) deployments we would like to design QKD solutions that provide for a target QKD key rate, in bits/pulse, at a specified upper-limit on the failure probability. However, in the finite-signalling setting, in which all real-world QKD systems exist, the common practice of achieving such a solution fails to deliver the maximum throughput rate of the classical decoder. This in turn means that the possibility that classical reconciliation becomes the bottleneck of the entire QKD protocol is not minimised. A design strategy that minimises this latter possibility, whilst achieving a target QKD rate with a target ceiling on the failure probability has not been developed – a situation we remedy here. Although our new design strategy detailed here is for LDPC codes and applied to two specific QKD protocols, the same strategy is generally applicable to all classical decoders and all QKD protocols. It is also deployable even in circumstances where the quantum bit error is variable, such as in satellite QKD systems.**

***Index Terms*— Quantum key distribution (QKD), LDPC codes, key reconciliation, satellite communications.**

I. Introduction

A SIGNIFICANT breakthrough in Quantum Key Distribution (QKD) was established in 2016 by the *Micius* Satellite in that, for the first time, a non-zero key rate was achieved over 1200 km – an order of magnitude improvement in distance relative to terrestrial deployments [1]. However, in the proof-of-principle QKD experiment by *Micius*, the sifted key is stored and processed after all the quantum signalling is complete. For real-time deployment of QKD in space we would like to consider the generation of key rates in real time: a process that involves classical post-processing whilst incoming quantum signals continue. In such a scenario, classical processing should occur at a rate faster than the incoming quantum signalling. Otherwise, the incoming quantum signals cannot be processed quickly enough and must be stored or discarded – an undesirable outcome in any pragmatic setting.

In the classical processing phase of QKD, the Key Reconciliation[1] (KR) is the slowest part of the process, and we focus on that element here. We assume that Low-Density Parity Check (LDPC) codes are utilised for the KR (although our strategies developed will be independent of that). We also ignore Privacy Amplification (PA), since for a given block length we can always find an efficient implementation of it that is faster than the KR [3].

In practical QKD deployments it would be useful to have a QKD solution that provides real-time generation of secure keys for a target QKD key rate (in bits/pulse[2]) with an upper limit for the protocol failure probability. Designing solutions that meet these two criteria poses a different challenge than that most often addressed in the literature – the maximisation of the bits/pulse QKD key rate [4]. However, maximisation of the bits/pulse rate fails to deliver the maximum LDPC decoding rate of the KR phase because no computational complexity at the KR phase is considered in this bits/pulse QKD rate. This in turn translates to a higher probability that KR becomes the bottleneck of the QKD protocol. A strategy that minimises this probability whilst achieving a target QKD key rate and target failure probability is not established yet. In this work we remedy this situation. Our new strategy points the way to real-world QKD deployments that deliver on pre-set specifications yet maximise the likelihood that no quantum signals will be dropped due to the overflow of the finite-size buffer.

Let us define ϵ as the total failure probability of a QKD protocol, f_t as the target QKD key rate of the QKD protocol (in bits/pulse), Q as the quantum bit error rate (QBER), Q_{tol} as the QBER tolerance, R_c as the LDPC code rate and C as the LDPC decoding rate of the KR (in bits per arithmetic operation). In this work, we devise a strategy that maximises C for a given ϵ, f_t, and QBER. We apply this strategy to two QKD protocols: the Asymmetric QKD protocol (A-QKD) [5], [6] and the Loss-Tolerant QKD (LT-QKD) protocol in [4], [7].

The rest of this letter is organized as follows: Section II introduces the QKD protocols considered in this letter. Section III describes the security of the A-QKD protocol in the finite-key regime and details the strategy for this protocol. Section IV analyses the LDPC decoding rate. Finally, Section V shows the design outcome for both protocols and discusses the viability of our designed system in the satellite-based scenario.

II. QKD Protocols Considered in This Letter

Unlike the well-known BB84 [8] protocol where Alice and Bob discard half of their measurement results during the key sifting, A-QKD allows Alice and Bob to discard less measurement results during the same process since the probability of choosing the X basis for measurement is higher relative to choosing the Z basis [5].The security of A-QKD is based on the assumption of a perfect quantum apparatus.

Manuscript received January 21, 2020; revised February 19, 2020; accepted February 21, 2020. Date of publication March 2, 2020; date of current version May 8, 2020. The work of Xiaoyu Ai is supported by the Research Training Program (RTP) Fee Offset and the Postgraduate Research Support Scheme (PRSS) at the University of New South Wales, Australia. The associate editor coordinating the review of this letter and approving it for publication was T. De Cola. *(Corresponding author: Xiaoyu Ai.)*

Xiaoyu Ai and Robert Malaney are with the School of Electrical Engineering and Telecommunications, University of New South Wales, Sydney, NSW 2052, Australia (e-mail: x.ai@unsw.edu.au).

Soon Xin Ng is with the School of Electronics and Computer Science, University of Southampton, Southampton SO17 1BJ, U.K.

Digital Object Identifier 10.1109/LCOMM.2020.2977914

[1]The KR is a processing phase of a QKD protocol where Alice and Bob's raw key strings are made identical with the help of a parity check matrix and a classical decoding algorithm [2].

[2]In many works, "QKD key rate" refers to the ratio of the number of secure key bits obtained (at the end of a QKD protocol) to the number of transmitted pulses – normally derived via a security analysis. However, we note that the computational complexity of the classical reconciliation phase is not considered when calculating this QKD key rate.

1558-2558 © 2020 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: UNIVERSITY OF SOUTHAMPTON. Downloaded on May 27,2020 at 22:03:52 UTC from IEEE Xplore. Restrictions apply.

The LT-QKD protocol, on the other hand, tries to close security loopholes raised by the imperfections of quantum devices. As an advanced BB84-variant, LT-QKD is "loss tolerant" in the sense that it is robust against the imperfection of photon generation devices, specifically phase modulation errors in the state preparation phase [7]. In [4], Mizutani *et al* provide an information-theoretic security analysis of the LT-QKD protocol in the finite-key regime.

III. Design Strategy

A. The Generic Design Strategy

In this section, we describe the generic steps of our design strategy with a given ϵ and f_t. We start with an initial R_c and then: (i) find the minimum block length where f_t is met; (ii) design a rate-R_c LDPC code; (iii) calculate C; (iv) reduce R_c and return to (i) until the maximum of C is found. We note that (i) is dependent on the designated QKD protocol and its security analysis in the finite-key regime. Actual implementation will vary somewhat from this generic design – dependent on the adopted QKD protocol. The detailed strategy for the A-QKD protocol is shown next.

B. Detailed Strategy for A-QKD

We adopt the finite-length security analysis of A-QKD in [6]. The A-QKD protocol is ϵ-secure if it is ϵ_{cor}-correct and ϵ_{sec}-secret with $\epsilon_{cor} + \epsilon_{sec} \leq \epsilon$ [6]. The protocol is ϵ_{cor}-correct if the probability that Alice's final key is not equal to Bob's is upper bounded by ϵ_{cor}. The protocol is ϵ_{sec}-secret if the joint system formed by Alice's final key and Eve's system is ϵ_{sec} close to the ideal system where Alice's key is uniformly distributed and Eve's system is not correlated to Alice's key when the protocol does not abort [6]. We assume ϵ_{cor} and ϵ_{sec} are set individually by the user (other strategies based on a given ϵ only are discussed later).

In the A-QKD protocol, a term that accounts for statistical fluctuations is introduced, namely, $\mu = \sqrt{\frac{(N+K)(K+1)}{NK^2} \log \frac{2}{\epsilon_{sec}}}$, where $K = N\left(1 - p_x^{-1}\right)^2$ is the number of samples used to estimate QBER, N is the length of the key of the input to the PA (the LDPC block length), and p_x is the probability of selecting the X basis during the state preparation [6]. For the given ϵ_{cor} and ϵ_{sec}, the security analysis of the A-QKD protocol in the finite-key regime can be used to calculate the upper bound of length of the final secure keys, l, as a function of Q_{tol} and N. Thus, assuming that the quantum states are prepared in the X or Z basis, the upper bound of the QKD rate, $f_s^A(Q_{tol}, N) \leq \frac{lp_x^2}{N}$, is obtained by [6]:

$$f_s^A(Q_{tol}, N) \leq p_x^2 \frac{1 - h(Q_{tol} + \mu) - N_{leak} - \log \frac{2}{\epsilon_{cor}\epsilon_{sec}^2}}{N}, \tag{1}$$

where $h(x) = -x \log x - (1-x)\log(1-x)$ is the binary entropy function and $N_{leak} = (1 - R_c)N$ is the number of bits disclosed in the KR. We assume here the bound given by Eq. (1) is tight [6], and for purposes of calculation set the inequality to an equality. However, strictly speaking the QKD rate in bits/pulse we use is only an upper bound on the target key rate.

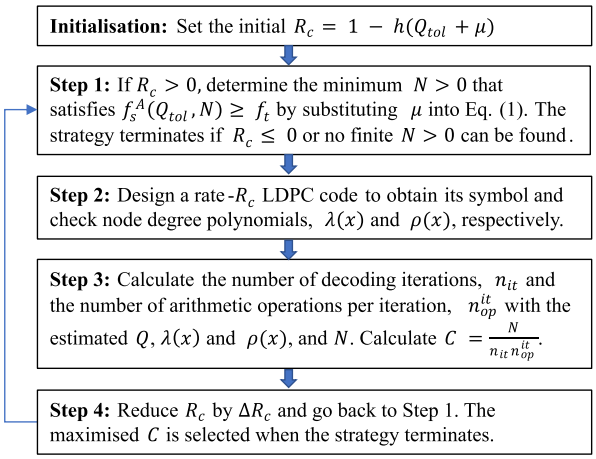

Fig. 1. A diagram showing all the steps of Strategy-I.

Detailing the generic steps of our strategy in Section III-A, we obtain the design strategy specifically for A-QKD (denoted by Strategy-I) for the user-defined ϵ_{sec}, and ϵ_{cor} (or ϵ). In the following we assume the QBER is fixed and known (we discuss later the satellite scenario, in which QBER is variable). Setting ϵ_{sec}, ϵ_{cor}, f_t, p_x and Q_{tol} (or ϵ, f_t, p_x and Q_{tol}) as inputs, $R_c = 1 - h(Q_{tol} + \mu)$ as the initial value, we describe Strategy-I in Fig. 1.

We note that LDPC code design is an optimisation problem with constraints: for a given QBER and a code rate, find each coefficient in the symbol and check degree polynomial, $\lambda(x)$ and $\rho(x)$, respectively, so that the decoding threshold is maximised. To solve this optimisation problem, the common approach is to apply Differential Evolution [9] as the numerical solver and the Density Evolution Algorithm (DEA) [10] to determine the threshold.

Our end result is that we have found, for a given channel represented by a QBER, the LDPC code that minimises the probability of KR being the bottleneck of the entire QKD protocol, whilst achieving a desired QKD key rate.

IV. LDPC Decoding Rate

In this section, we illustrate how the complexity analysis of the LDPC decoding is used to calculate n_{it} and n_{op}^{it} in Step 3 of Strategy-I for the given N, $\lambda(x) = \sum_{i=2}^{\overline{d_\lambda}} \lambda_i x^{i-1}$ and $\rho(x) = \sum_{i=2}^{\overline{d_\rho}} \rho_i x^{i-1}$ where $\overline{d_\lambda}$ and $\overline{d_\rho}$ are the maximum symbol and check node degree, respectively. It is shown in [11] that for the binary symmetric channel (BSC), n_{it} is a decreasing function of the distance between a given rate-R_c LDPC code and the capacity at the given QBER [12]. Taking the trade-off between n_{it} and this distance into account, we can improve C by reducing R_c at Step 4 of Strategy-I.

A. The Number of Decoding Iterations

n_{it} is the number of decoding iterations that is executed by the decoder to reach a given decoding error, ϵ_{cor}. Generally, n_{it} can be analysed by applying the DEA over the BSC based on the assumption of a cycle-free Tanner Graph. However, in practice, LDPC codes with finite size always contain cycles [13]. As QBER increases to a transition value Q_{th},

Authorized licensed use limited to: UNIVERSITY OF SOUTHAMPTON. Downloaded on May 27,2020 at 22:03:52 UTC from IEEE Xplore. Restrictions apply.

messages propagated in a finite-size Tanner Graph start to circulate at a certain iteration number. We also calculate n_{it} from the girth of the LDPC Tanner Graph constructed by a specific construction algorithm. Therefore, we obtain n_{it} by considering two cases:

- Case 1: we obtain the number of decoding iterations from the DEA, n_{it}^{DEA} where $0 \leq Q \leq Q_{th}$;
- Case 2: we obtain the number of decoding iterations with the girth analysis of the Progressive Edge Growth (PEG) algorithm [13], n_{it}^{PEG} where $Q \geq Q_{th}$.

Note that Q_{th} is the value of the QBER when $n_{it}^{\text{DEA}} = n_{it}^{\text{PEG}}$.

In Case 1, we adopt the DEA for the serial-schedule Belief Propagation (BP) decoder because this decoding algorithm requires less iterations relative to the flooding-schedule BP decoder [14]. For a given QBER and an LDPC code defined by the degree distribution pair, $\lambda(x)$ and $\rho(x)$, the decoding error after the j^{th} iteration, p_j, is a monotonously decreasing function of j [14]. Therefore, we analyse n_{it}^{DEA} by using the following equation for a given QBER:

$$n_{it}^{\text{DEA}} = \arg\min_{j}\{p_j = f(Q, j, \lambda(x), \rho(x)) \leq \epsilon_{cor}, j \in \mathbb{N}\}, \tag{2}$$

where $f(Q, j, \lambda(x), \rho(x))$ is the recursive function of the DEA for the serial-schedule BP decoder. Now we show the explicit form of $f(Q, j, \lambda(x), \rho(x))$ in the BSC. It is shown that p_j is obtained by [14]:

$$p_j = \int_{-\infty}^{0} \Omega_p^0 dx = \int_{-\infty}^{0} \big\{P_{L|C}(u,j)\Omega_p^{2u} + (1 - P_{L|C}(u,j)) \Omega_p^{2u} * \lambda(\Gamma^{-1}(\rho(\Gamma(\Omega_p^{2u+2}))))\big\}\, dx, \tag{3}$$

where $P_{L|C}(u,j), u = j-1, \cdots, 1, 0$ is the probability that a symbol node is recognized as a leaf node in a computation tree at Tier $2j$ (calculated by Proposition III-5 in [14]), Ω_p^j is the density of the decoding error at the j^{th} iteration, $\Gamma(x)$ and its inverse $\Gamma^{-1}(x)$ represent the two transforms of the probability mass function defined in [10], and the operator $*$ represents the discrete convolution. The initial density is given by $\Omega_p^{2u} = Q\delta(x - \log\frac{Q}{1-Q}) + (1-Q)\delta(x + \log\frac{Q}{1-Q})$, where $\delta(x)$ is the Dirac Delta Function [10]. For BSC, Eq. (3) can be rewritten as follows [14], [15]:

$$p_j = QP_{L|C} - (1 - P_{L|C})\Big[\sum_{k=1}^{\overline{d_\lambda}} \lambda_k \big[Qg(\rho(1-2p_{2u+2}), k, b) + (1-Q)\, g(-\rho(1-2p_{2u+2}), k, b)\big]\Big] \tag{4}$$

where b is given in Section III-B in [15] and $g(x)$ is given by [15]:

$$g(x, k, b) = \sum_{t=b}^{k-1} \binom{k-1}{t} \left(\frac{1+x}{2}\right)^t \left(\frac{1-x}{2}\right)^{k-1-t}. \tag{5}$$

For Case 2, we consider that the parity check matrix used in the KR is constructed by the PEG algorithm [13] for the given degree distributions. We first relate n_{it}^{PEG} to the girth, G, of the LDPC parity check matrix, and n_{it}^{PEG} can be upper bounded by $n_{it}^{\text{PEG}} < \frac{G}{4}$, where G is obtained by $G = 2(\lfloor a \rfloor + 2)$, and a is obtained by [13]: $a = \frac{\log((1-R_c)N\overline{d_\rho} - \frac{(1-R_c)N\overline{d_\rho}}{d_\lambda - (1-R_c)N+1})}{\log(\overline{d_\rho}-1)(\overline{d_\lambda}-1)} - 1$.

Putting the two cases together, n_{it} is obtained by using the following equation:

$$n_{it}(Q, N) = \begin{cases} n_{it}^{\text{DEA}}, & Q \leq Q_{th} \\ n_{it}^{\text{PEG}}, & Q > Q_{th}, \end{cases} \tag{6}$$

where Q_{th} is equal to Q when $n_{it}^{\text{DEA}} = n_{it}^{\text{PEG}}$.

B. The Number of Arithmetic Operations

Based on the pseudo-code (Table 1 of [16]) of the serial-schedule BP decoder, we analyse n_{op}^{it} for the given $\lambda(x)$, $\rho(x)$, and N by counting the number of floating point arithmetic operations involved in each step of the algorithm within one decoding iteration. We note that the Log-Likelihood Ratio (LLR) messages with value x are transformed between $\mathbb{R}$ and $\{-1, 1\} \times \mathbb{R}$ by $\phi(x) = \{\text{sign}(x), -\log\tanh(\frac{|x|}{2})\}$ and its inverse $\phi^{-1}(\text{sign}, x) = ((-1) \cdot \text{sign}) \log\tanh(\frac{|x|}{2})$. The addition (subtraction) that is defined in the domain, $\{-1, 1\} \times \mathbb{R}$, costs one floating point and one bit-wise operation. We omit the computational cost of the bit-wise operation and only consider the floating-point operations here. The detailed numbers of operations for different arithmetic operations are: $4M(\sum_{i=2}^{\overline{d_\rho}} i\rho_i)$ additions, $2M(\sum_{i=2}^{\overline{d_\rho}} i\rho_i)$ calculations of $\phi(x)$ and $M(\sum_{i=2}^{\overline{d_\rho}} i\rho_i)$ calculations of $\phi^{-1}(x)$ where $M = (1 - R_c)N$ is the number of rows in an LDPC parity check matrix. Therefore, the n_{op}^{it} can be determined by summing the number of floating point operations together: $n_{op}^{it} = 7(1 - R_c)N(\sum_{i=2}^{\overline{d_\rho}} i\rho_i)$.

V. Simulation Results

A. The A-QKD Protocol

The results obtained from the use of Strategy-I for the A-QKD are shown (red curves) in Fig. 2 for $Q \in [0.01, 0.1]$. We set $\Delta R = 0.1$. In our simulations we have set Q_{tol} to 0.01 higher than the true QBER. This is a pragmatic choice and previous optimisation studies have shown this to be a setting (averaged over a range of QBER) that maximises the key rate for a set security setting [6]. Due to the fact that n_{it} is a non-decreasing function of the QBER, the C obtained from Strategy-I under this assumption is the worst case scenario. To obtain the results shown we apply Strategy-I at each QBER to ensure that C at each QBER is maximised for each ϵ and f_t. We see that for a given $\epsilon = \epsilon_{cor} + \epsilon_{sec}$ and f_t, the general trend is that C decreases as the QBER increases. Not shown in Fig. 2 is that the R_c obtained by the strategy (the minimum R_c at each QBER) is a monotonically decreasing function of the QBER. For a low QBER, the C is higher because the obtained R_c is found significantly below capacity (resulting in significantly lower n_{it}) while f_t is still met. However, for a high QBER, the C is penalized because the obtained R_c is close to capacity (a higher R_c is needed so as to ensure that a finite $N > 0$ consistent with the f_t is found).

In Fig 2, we also plot the LDPC decoding rate for a widely used rate-adaptive reconciliation scheme [2], hereby referred as the RRP (Rate-adaptive Reconciliation using Puncturing) scheme. For each estimated QBER, this scheme adapts an LDPC code via puncturing and shortening so that amount of

Authorized licensed use limited to: UNIVERSITY OF SOUTHAMPTON. Downloaded on May 27,2020 at 22:03:52 UTC from IEEE Xplore. Restrictions apply.

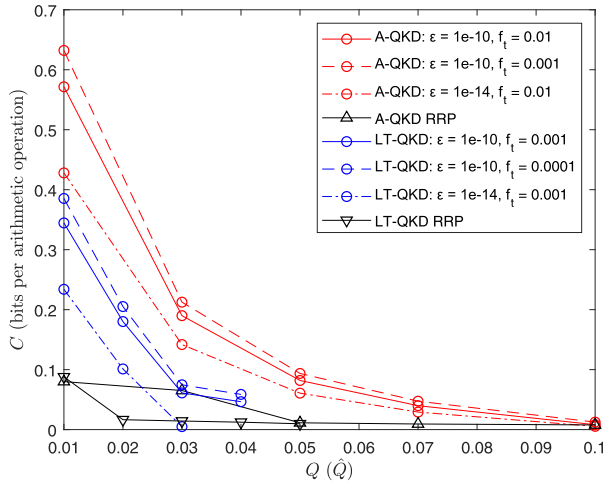

Fig. 2. The LDPC decoding rate C (in bits per arithmetic operation) vs. Q ($\hat{Q}$) for the ϵ-secure A-QKD (LT-QKD) protocol for different target QKD rate. For A-QKD, we consider $f_t = 0.01$ and $f_t = 0.001$ for $\epsilon_{cor} = 9 \times 10^{-11}$ ($\epsilon_{cor} = 9 \times 10^{-15}$) and $\epsilon_{est} = 10^{-11}$ ($\epsilon_{est} = 10^{-15}$), respectively. p_x is set to 0.75. As for the LT-QKD, we consider the following parameter settings: $N_{tag} = 0$ and $p_{fail} = 0$; For $\epsilon = 10^{-10}$ (10^{-14}), $\epsilon_{cor} = \epsilon_s = \frac{\epsilon}{2}$, $\epsilon_z = \frac{\epsilon_{cor}}{2}$, $\epsilon_{pa} = \epsilon_{ph} = \frac{\epsilon_s^2}{16}$, and $p_Z^A = p_Z^B = 0.8$.

the syndrome bits disclosed at the KR phase can be minimised. The LDPC decoding rate obtained using the RRP scheme for the A-QKD protocol is shown in Fig. 2, where we have set $\epsilon = 10^{-10}$ and $f_t = 0.01$. In Fig. 2, we observe that our strategy leads to higher C relative to the RRP scheme.

In Fig. 2, for a given QBER, in the case where we decrease ϵ and/or increase f_t, C is penalized by the increasing n_{it} because the obtained R_c is closer to the capacity at this QBER. We also find that N is increasing in the same case but the increasing N does not affect C because n_{op}^{it} also scales as $O(N)$. We can easily convert C (in bits per arithmetic operation) in Fig. 2 into a bits/second (BPS) rate via $\frac{C}{t_p}$, where t_p is the hardware-dependent time required for one arithmetic operation.[3]

We note that the $\frac{C}{t_p}$ calculated is an averaged BPS rate over the timescale, t_d, of LDPC decoding. In the cases where N is large, t_d can be large.[4] Typical values of N found for A-QKD are shown in Table I for various security settings. Clearly, in any operational setting for real-time QKD we would want t_d to be set below a well-defined threshold timescale, t_{th}.

The above discussion leads us to our definition of a viable QKD system. For an ϵ-secure QKD protocol, we define a *viable* system to be a system for which (i) $\frac{C}{t_p}$ and the incoming sifted key rate (in BPS), R_s, satisfy $\frac{C}{t_p} \geq R_s$ for all QBER that delivers non-zero QKD key rate; and (ii) $t_d < t_{th}$.

B. Other QKD Protocols

Although Strategy-I is based on the A-QKD protocol, this strategy can be applied to other QKD protocols with minor changes. As an example, we consider the LT-QKD protocol [4] mentioned in the introduction and devise a strategy for this protocol via a simple change to Step 1 of Strategy-I.

To briefly describe the security analysis for the LT-QKD protocol from [4], let us first define ϵ_{pa} as the collision probability of the 2-universal hash function used in the PA (embedded in ϵ_{sec} in A-QKD), E_{det} as the number of phase errors in single-photon detection events using the Z basis, N_{det} as the total number of single-photon detection events using the Z basis, E_{det}^U as the upper bound of the confidence interval when estimating E_{det}, and N_{det}^L as the lower bound of the confidence interval when estimating N_{det}, ϵ_{ph} as the upper bound of the probability that the estimated E_{det} is larger than E_{det}^U, and ϵ_z as the upper bound of the probability that N_{det} is less than N_{det}^L.

In the parameter estimation step, instead of directly estimating the QBER from the testing set as in A-QKD, Alice needs to estimate E_{det} and N_{det} separately. This leads to the two new parameters in the security analysis – ϵ_{ph} and ϵ_z. The LT-QKD protocol described in [4] is ϵ-secure if it is ϵ_{cor}-correct and ϵ_{sec}-secret with $\epsilon = \epsilon_{cor} + \epsilon_{sec} = \epsilon_{cor} + \sqrt{2(\epsilon_{pa} + \epsilon_{ph})} + \epsilon_z$. Thus, for the user-defined ϵ_z, ϵ_{pa}, ϵ_{ph}, and ϵ_{cor}, the QKD rate of the LT-QKD, $f_s^L(\hat{Q}, N)$, is upper bounded by [4]:

$$f_s^L(\hat{Q}, N) \leq \frac{p_d \left(N_{det}^L \left(1 - h\left(\hat{Q}\right)\right) - N_{leak} - \log_2 \frac{2}{\epsilon_{cor}\epsilon_{pa}} \right)}{N} \tag{7}$$

where $\hat{Q} = \frac{E_{det}^U(\epsilon_{ph})}{N_{det}^L(\epsilon_z)}$ is related to the estimated QBER from the quantum channel (see [4] for exact definition), and p_d is the probability that Alice and Bob select the same basis for the quantum measurement. In [4], $p_d = p_Z^A p_Z^B$, where p_Z^A (p_Z^B) is the probability of Alice (Bob) selecting the Z basis for the quantum measurement. N_{det}^L and E_{det}^U are calculated by Eq. 21 and 24 in [4].

We describe the revised strategy designed specifically for the LT-QKD protocol for the user-defined ϵ_z, ϵ_{pa}, ϵ_{ph}, and ϵ_{cor}. Compared to the A-QKD protocol, the estimation of QBER in the LT-QKD protocol is more complex because both E_{det} and N_{det} need to be estimated with the confidence levels associated with the given ϵ_{ph} and ϵ_{ph}, respectively. Setting ϵ_z, ϵ_{pa}, ϵ_{ph}, ϵ_{cor}, f_t, p_Z^A, p_Z^B and $\hat{Q}$ (or ϵ, f_t, p_Z^A, p_Z^B and $\hat{Q}$) as inputs, $R_c = 1 - h(\hat{Q})$ as the initial value, we obtain this revised strategy by replacing Step 1 of Strategy-I with the following: If $R_c > 0$, determine the minimum $N > 0$ that satisfies $f_s^L(\hat{Q}, N) \geq f_t$ using Eq. (7); The strategy terminates if $R_c \leq 0$ or no finite $N > 0$ can be found.

The results of this revised strategy for the LT-QKD protocol are shown (blue curves) in Fig. 1. The general trend found for LT-QKD is the same as the trend for A-QKD, albeit at lower LDPC decoding rates C. We also find that the RRP scheme still leads to lower C compared to our strategy. However, we observe that C goes to zero for LT-QKD. This is due to the fact that no finite N, which satisfies $f_s^L(\hat{Q}, N) \geq f_t$, can be found for $\hat{Q} > 0.04$ ($\hat{Q} > 0.03$ when $\epsilon = 10^{-14}$). The values of N for different security settings for the LT-QKD protocol are given in Table I.

[3]For example, for a HP Z8 G4 workstation we find $t_p = 8 \times 10^{-8}$ seconds. This in turn means $\frac{C}{t_p}$ equals 5.12 Mbps, 0.97 Mbps, at $Q = 0.01$ and 0.05, respectively.

[4]For example, using $t_p = 8 \times 10^{-8}$ seconds we find that $t_d \sim 40$ seconds when $N = 10^8$.

Authorized licensed use limited to: UNIVERSITY OF SOUTHAMPTON. Downloaded on May 27,2020 at 22:03:52 UTC from IEEE Xplore. Restrictions apply.

TABLE I

TYPICAL VALUES OF N (IN bits) FOR DIFFERENT ϵ, f_t AND Q ($\hat{Q}$)

$f_t = 10^{-3}$				
$Q(\hat{Q})$	0.01		0.05	
ϵ	10^{-10}	10^{-14}	10^{-10}	10^{-14}
N (A-QKD)	8.2×10^3	2.2×10^4	2.8×10^4	8.7×10^4
N (LT-QKD)	2.4×10^6	4.7×10^8	–	–
$f_t = 10^{-4}$				
$Q(\hat{Q})$	0.01		0.05	
ϵ	10^{-10}	10^{-14}	10^{-10}	10^{-14}
N (A-QKD)	7.3×10^3	6.1×10^4	2.1×10^4	2.9×10^4
N (LT-QKD)	1.9×10^6	3.8×10^8	–	–

C. Satellite-Based QKD Protocols

We now return to the discussion of satellite-based QKD. In satellite-based QKD, the QBER is very much a time-dependent quantity [1]. If our strategy is to be applicable to such a setting, we must repeat it for every QBER that is estimated. In doing this we note that the coherence timescale of the satellite-to-ground channel, is of order 10-100 ms [1] – a timescale significantly longer than the time to reload a processing system with a new LDPC code retrieved from on-board memory [17].

This fast reloading of codes allows us to adapt our strategy for space-based deployments in the following manner. We *a-priori* build a look-up table consisting of a set of LDPC codes which maximise $\frac{C}{t_p}$ at the given QBER in the anticipated range of the satellite-to-ground channel, typically $0 - 0.1$ (above this range QKD rates approach zero for other reasons [6]). Upon estimating the QBER at any instant in time, the LDPC code that maximises $\frac{C}{t_p}$ for that QBER is selected.

For most satellite deployments our system will be viable (defined in Section V-A) for a wide range of parameter settings. For example, in a variable channel consistent with the range of QBER reported by *Micius* [1], we find that for a failure probability bound of $\epsilon = 10^{-10}$ and key rate $f_t = 0.01$, the incoming quantum signal rate would have to exceed $10.2c$ ($7.5c$) (Mbps) in order for the LDPC decoder to become the bottleneck in real-time deployments of the A-QKD (or LT-QKD) protocol, where $c = \frac{8 \times 10^{-8}}{t_p}$.

D. Other Design Strategies

Our design strategies for the A-QKD and LT-QKD protocols can also be applied to several modified input and output requirements. For example, in A-QKD a user can instead set a value of ϵ, with the values of ϵ_{cor} and ϵ_{sec} then obtained while maximizing the QKD key rate under the constraint $\epsilon_{cor} + \epsilon_{sec} \leq \epsilon$ and a given QBER [6]. Note that this maximization leads to extra computations. That is, we would jointly find the optimal ϵ_{sec} and ϵ_{cor} that maximises $f_s^A(Q_{tol}, N)$ for all N. Then, we would search N within an *a-priori* defined interval, $[0, 10^{14}]$ and find the minimum N that satisfies $f_s^A(Q_{tol}, N) \geq f_t$.

Similarly for LT-QKD users could also set ϵ and find the optimal values of ϵ_z, ϵ_{pa}, ϵ_{ph}, and ϵ_{cor} that maximise the QKD key rate. Explicitly, if ϵ was used as the input, we would jointly find the optimal ϵ_z, ϵ_{pa}, ϵ_{ph}, ϵ_{cor} that maximise $f_s^L(\hat{Q}, N)$ for all N. Then, we would search N within an *a-priori* defined interval, $[0, 10^{14}]$ and find the minimum N that satisfies $f_s^L(\hat{Q}, N) \geq f_t$.

VI. CONCLUSION

In this work we have devised a design strategy where a quantum key distribution system achieves a target bits/pulse key rate with a specified total failure probability whilst maximising the bits-per-second rate of the key reconciliation phase. This strategy generally holds for various protocols, different classical decoding algorithms, and can be applicable to the time-varying quantum channel in satellite-based situations. Variants of our strategy can be applied to different design criteria resulting in a plethora of strategies all with the common thread that the design outcomes are conditioned on the decoding rate being maximised. Such an outcome minimises the probability that quantum signals will be discarded. We expect that our strategies point the way to the real-world deployment of practical quantum key distribution systems where all quantum signals can be utilised in the key growth with maximised probability.

REFERENCES

[1] S.-K. Liao *et al.*, "Satellite-to-ground quantum key distribution," *Nature*, vol. 549, p. 43, Aug. 2017.

[2] D. Elkouss, J. Martinez, D. Lancho, and V. Martin, "Rate compatible protocol for information reconciliation: An application to QKD," in *Proc. IEEE Inf. Theory Workshop (ITW)*, Jan. 2010, pp. 1–5.

[3] Z. Yuan *et al.*, "10-Mb/s quantum key distribution," *J. Lightw. Technol.*, vol. 36, no. 16, pp. 3427–3433, Aug. 15, 2018.

[4] A. Mizutani *et al.*, "Quantum key distribution with setting-choice-independently correlated light sources," *npj Quantum Inf.*, vol. 5, p. 8, Jan. 2019.

[5] H.-K. Lo, H. F. Chau, and M. Ardehali, "Efficient quantum key distribution scheme and a proof of its unconditional security," *J. Cryptol.*, vol. 18, pp. 133–165, Apr. 2005.

[6] M. Tomamichel, C. C. W. Lim, N. Gisin, and R. Renner, "Tight finite-key analysis for quantum cryptography," *Nature Commun.*, vol. 3, no. 1, p. 634, Jan. 2012.

[7] K. Tamaki, M. Curty, G. Kato, H.-K. Lo, and K. Azuma, "Loss-tolerant quantum cryptography with imperfect sources," *Phys. Rev. A, Gen. Phys.*, vol. 90, no. 5, Nov. 2014, Art. no. 052314.

[8] C. H. Bennett and G. Brassard, "Quantum cryptography: Public key distribution and coin tossing," in *Proc. IEEE Int. Conf. Comput., Syst. Signal Process.*, vol. 175, Dec. 1984, p. 8.

[9] R. Storn and K. Price, "Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces," *J. Global Optim.*, vol. 11, pp. 341–359, Dec. 1997.

[10] T. J. Richardson, M. A. Shokrollahi, and R. L. Urbanke, "Design of capacity-approaching irregular low-density parity-check codes," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 619–637, Feb. 2001.

[11] B. Smith, M. Ardakani, W. Yu, and F. Kschischang, "Design of irregular LDPC codes with optimized performance-complexity tradeoff," *IEEE Trans. Commun.*, vol. 58, no. 2, pp. 489–499, Feb. 2010.

[12] I. Sason and R. Urbanke, "Parity-check density versus performance of binary linear block codes over memoryless symmetric channels," *IEEE Trans. Inf. Theory*, vol. 49, no. 7, pp. 1611–1635, Jul. 2003.

[13] X.-Y. Hu, E. Eleftheriou, and D. M. Arnold, "Regular and irregular progressive edge-growth tanner graphs," *IEEE Trans. Inf. Theory*, vol. 51, no. 1, pp. 386–398, Jan. 2005.

[14] E. Sharon, S. Litsyn, and J. Goldberger, "Efficient serial message-passing schedules for LDPC decoding," *IEEE Trans. Inf. Theory*, vol. 53, no. 11, pp. 4076–4091, Nov. 2007.

[15] M. G. Luby, M. Mitzenmacher, M. A. Shokrollahi, and D. A. Spielman, "Improved low-density parity-check codes using irregular graphs," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 585–598, Feb. 2001.

[16] E. Sharon, S. Litsyn, and J. Goldberger, "An efficient message-passing schedule for LDPC decoding," in *Proc. 23rd IEEE Conv. Electr. Electron. Eng.*, Sep. 2004, pp. 223–226.

[17] S. Sharifi Tehrani, S. Mannor, and W. J. Gross, "Fully parallel stochastic LDPC decoders," *IEEE Trans. Signal Process.*, vol. 56, no. 11, pp. 5692–5703, Nov. 2008.

Authorized licensed use limited to: UNIVERSITY OF SOUTHAMPTON. Downloaded on May 27,2020 at 22:03:52 UTC from IEEE Xplore. Restrictions apply.