秘密鍵系列一致プロトコル並列型Cascadeの最適化

Optimization of Secret-Key Reconciliation Protocol Cascade with Parallel Procedure

小田戸毅*, 山崎浩一*

Takeshi Odato* and Koichi Yamazaki *

*玉川大学工学部ソフトウェアサイエンス学科, 194-8610 東京都町田市玉川学園6-1-1

*Department of Software Science, College of Engineering, Tamagawa University,

6-1-1 Tamagawagakuen Machida-shi Tokyo 194-8610

Abstract

Secret-key reconciliation protocol “Cascade” with parallel procedure is optimized based on some properties of the protocol obtained by computer simulations. By the proposed optimization, the number of bits announced over the public channel during the procedure decreases by about 10% at the sacrifice of increase in the number of round communication.

Keywords: secret-key reconciliation, error control code, quantum key distribution

1. はじめに

近年，暗号技術が著しい発展を遂げている．特に，量子暗号鍵配送(BB84)プロトコル[1]は，様々な機関で実用実験が行われている．BB84プロトコルでは，暗号通信に用いる系列(暗号鍵)を第三者に知られることなく，正規の送受信者間で共有することが可能である．しかし，現実的なシステムでは，量子通信路の不完全性や雑音などにより，送受信者が共有した乱数系列の間に不一致が生じる．

秘密鍵系列一致プロトコルは，不一致を有する送受信者の乱数系列を一致させるためのプロトコルである．秘密鍵系列一致プロトコルは，対話型プロトコルと非対話型プロトコルに分類される．対話型プロトコルは非対話型プロトコルと比較して，広い範囲の誤り率において系列を一致させることが可能である[2]．したがって，幅広い誤り率の量子通信路を用いて鍵系列の共有を行う場合，対話型秘密鍵系列一致プロトコルが有用である．

対話型秘密鍵系列一致プロトコルとして，BBBSSプロトコル[3]やCascadeプロトコル[4]が知られている．BBBSSプロトコルは，最初に行われた量子暗号鍵配送の実験で用いられたプロトコルである．このプロトコルの誤り訂正能力を改善すべくCascadeプロトコルは提案された．Cascadeプロトコルは，シャノン限界に非常に近いbit数を公開することで，系列の一致処理を行うことが可能である．一方，鍵系列の共有を行う伝送路の誤り率の増加に伴い一致処理に必要な通信回数が増えてしまう問題を有する[5]．そこで，通信回数の削減を目的に並列型Cascade[6]が提案された．並列型Cascadeは，通信回数を大幅に削減することが可能だが，公開するbit数がわずかに増加してし

まう．

本研究では，計算機シミュレーションにより並列型Cascadeの特性を明らかにし，その特性に基づき公開するbit数を削減することを目的としたプロトコルの最適化手法を提案する．

2. 従来型Cascadeプロトコル[4)]

従来型Cascadeプロトコルは，Brassard, Salvailらによって提案された対話型秘密鍵系列一致プロトコルである．passと呼ばれる複数のステップで構成されており，シャノン限界に近いbit数で不一致の解消が可能である．以下に，従来型Cascadeのアルゴリズムを示す．

正規の送信者Alice，受信者Bobが量子通信路を用いて共有したそれぞれの鍵系列を$A=(a_1,\dots,a_n), B=(b_1,\dots,b_n)$とする．共有する鍵系列は二元乱数系列$(\{0,1\}^n)$である．

BBBSSプロトコルやCascadeプロトコルでは，訂正処理にBINARYを用いる．まず，BINARYのアルゴリズムについて説明する．AliceとBobがBINARYを施す系列には，奇数個の誤りが存在するものと仮定する．

1. AliceとBobは，BINARYを施す対象の系列を半分に分割し，系列の前半・後半部分のパリティを導出する．
2. パリティの比較を行う．パリティの異なる系列を半分に分割し，同様にパリティを導出する．

この処理を誤りが特定できるまで再帰的に行う．一回のBINARYで１つの誤りを訂正することができる．続いて，Cascadeのアルゴリズムを示す．

pass1：

1. AliceとBobはブロックサイズk_1を定め，鍵系列を$k_1$bitのブロックに分割する．pass1において，l番目のbitがv番目のブロックに含まれる場合，$K_v^1=\{l|(v-1)k_1\le l<vk_1\}$と表記する．
2. ブロック毎にパリティを算出し，公衆通信路を用いてAlice，Bob間で比較を行う．
3. パリティの異なるブロック全てに対して，BINARYを用いることで，ブロック毎に1つの誤りを訂正する．

pass1で使用した系列はpass1以降でも使用するため，系列を保持する．

pass$i$ $(i>1)$：

1. AliceとBobは，pass$i$ で用いるブロックサイズk_iを定める．
2. ランダム関数$f_i:[1,\dots,n]\to[1,\dots,\lceil n/k_i\rceil]$を選択し，系列の並びをスクランブルし，指定したサイズのブロックに分割する．pass$i$において，j番目のブロックにl番目のbitが含まれる場合，$K_j^i=\{l|f_i(l)=j\}$と表記する．
3. ブロック毎にパリティを求め，公衆通信路を用いてAlice，Bob間で比較を行う．
4. パリティの異なるブロックのうち，最もブロックサイズが小さいブロック1つに対して，BINARYを施すことで1つの誤りを訂正する．

4の処理を，$\mathcal{K}'=\phi$ を満たすまで再帰的に行う．ただし，

$$\mathcal{K}'=\mathcal{B}\nabla\mathcal{K} \qquad (1)$$

である．式(1)は，差集合$\mathcal{B}\nabla\mathcal{K}=(\mathcal{B}\cup\mathcal{K})\setminus(\mathcal{B}\cap\mathcal{K})$である．また，$\mathcal{K}$は異なるブロックパリティを持つブロック全体，$\mathcal{B}$はBINARYが施されたブロックの集合である．

3. 並列型Cascadeプロトコル[6)]

並列型Cascadeは，従来型Cascadeの通信回数を削減するために提案された対話型秘密鍵系列一致プロトコルである．

pass1：

従来型Cascadeの処理と同様に行う．

pass$i$ $(i>1)$：

1. AliceとBobは，pass$i$で用いるブロックサイズk_iを定める．
2. ランダム関数$f_i:[1,\dots,n]\to[1,\dots,\lceil n/k_i\rceil]$を選択し，系列の並びをスクランブルし，指定し

たサイズのブロックに分割する．

3. ブロック毎にパリティを導出し，公衆通信路を用いてAlice，Bob間で比較を行う．
4. パリティの異なるブロックのうち，最もブロックサイズが小さいブロック“すべて”に対してBINARYを施すことで，それぞれのブロックにおいて一つの誤りを訂正する．

プロトコルの終了条件は，従来型Cascadeと同様とする．

4. 評価基準

対話型秘密鍵系列一致プロトコルの性能評価には，公開ビット数と通信回数を用いる．公開ビット数の評価には以下の式で与えられるEfficiencyを用いる．

$$f_{\mathrm{EC}} = \frac{m}{nH(A|B)} \qquad (2)$$

ここで，mは訂正処理において公開されたビット数，nは共有する鍵系列の長さである．誤り率εの2元対称通信路を用いると仮定すると以下のように表現することができる．

$$f_{\mathrm{EC}} = \frac{1-R}{h(\varepsilon)} \qquad (3)$$

ここで，$h(\varepsilon)$は2値エントロピー，$R = 1 - m/n$である．通信回数については，Alice，Bob間の一往復の通信をround communicationとしてその回数で評価を行う．

5. 計算機シミュレーション

Efficiencyとround communicationについて，従来型Cascadeと並列型Cascadeの計算機シミュレーションを行なった．誤り率を0.0001から0.15まで0.0001ずつ変動させ，各誤り率に対して10,000回の計算機シミュレーションを行なった．ブロックサイズは以下の2式を満たす最大の整数を用いる[4]．

$$\sum_{l=j+1}^{\lfloor k_1/2 \rfloor} \delta_1(l) \le \frac{1}{4}\delta_1(j) \qquad (4)$$

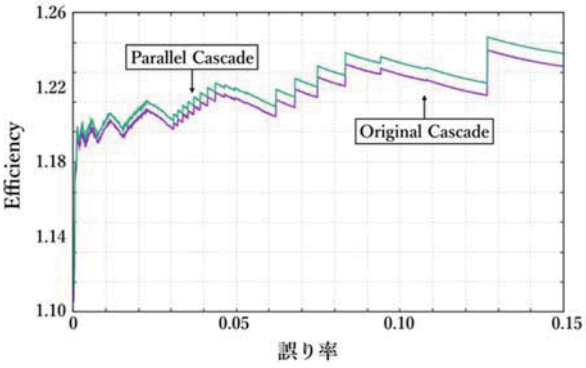

図1 Efficiency(従来型，並列型)

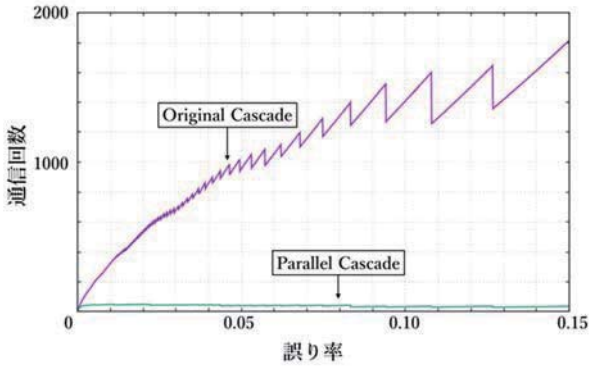

図2 通信回数(従来型，並列型)

$$1 - (1 - e^{-2E_1})^2 \ge \frac{3}{4} \qquad (5)$$

$\delta_1(j)$はpass1終了後に，あるブロックが$2j$個の誤りを含む確率であり，E_1はpass1終了時にあるブロックに含まれる誤りの個数の期待値であり，次式で与えられる．

$$E_1 = k_1\varepsilon - \frac{(1-(1-2\varepsilon)^{k_1})}{2} \qquad (6)$$

ただし，ε は量子通信路の誤り率である．pass$i$ (>1)のブロックサイズk_iは，$k_i = 2k_{i-1}$とする．

図1にEfficiencyを示す．従来型Cascadeと比較して並列型Cascadeの公開ビット数は1%程度増加しており，Efficiencyが劣化していることが確認できる．

図2に通信回数を示す．従来型Cascadeと比較して，並列型Cascadeでは大幅にround communicationが削減されていることが確認できる．従来型Cascadeが，誤り率の増加に伴いround communicationが増えているのに対して，並列型Cascadeでは誤り率によらずほぼ一定であることが分かる．

また，各pass終了時の誤り率の推移についての検討を以下で行う．まず，並列型Cascadeの各pass終了後の誤り率を表1に示す．表1より，pass2終了時点でほとんど全ての誤りが訂正されることが確認できる．次に，並列型Cascadeにおいて，誤りがどの系列で訂正されるかに関する割合を表2に示す．表2より，pass1の系列で70%弱の誤りが訂正され，pass2の系列で残りの約30%が訂正されていることが確認できる．

6. 並列型Cascadeの最適化

計算機シミュレーションの結果より，並列型CascadeのEfficiencyは従来型Cascadeと比較して劣化していることが明らかとなった．そこで，本節では並列型CascadeのEfficiencyの向上を目的としたブロックサイズの最適化法を示す．

表1,2から得られた計算機シミュレーションから以下の2つの性質を仮定する．

1. pass2終了時にほとんど全ての誤りが訂正される．
2. pass1系列で誤り全体のRが訂正され，残りの誤りがpass2系列で訂正される．

ここで，Rを任意の割合とする．

以上の二つの性質を仮定すると，訂正処理で公開されるビット数の期待値は

$$n_{\mathrm{bit}}^{\mathrm{Opt.}} = \frac{n}{k_1} + Rn\varepsilon \log_2 k_1 + \frac{n}{k_2} + (1-R)n\varepsilon \log_2 k_2 \quad (7)$$

で与えられる．ここで，nは共有する鍵系列の長さである．式(6)から解析的な手法により，公開ビット数を最小にするようなブロックサイズは次式で与えられる．

$$k_1 = \left\lfloor \frac{\ln 2}{R\varepsilon} \right\rfloor, k_2 = \left\lfloor \frac{\ln 2}{(1-R)\varepsilon} \right\rfloor \quad (8)$$

以上で得られた結果に対して，計算機シミュレーションの結果に基づき，割合Rを2つの場合で仮定してブロックサイズの最適化を行う．Opt.1では$R = 2/3$，Opt.2では$R = 7/10$とする．

表1 各pass終了後の誤り率

ε	pass1	pass2	pass3	pass4
0.01	0.004725	0.000099	0.000002	0.000000
0.05	0.022438	0.000086	0.000000	0.000000
0.10	0.043546	0.000077	0.000000	0.000000
0.15	0.066841	0.000082	0.000000	0.000000

表2 各系列で訂正される誤りの割合

ε	pass1	pass2	pass3	pass4
0.01	0.670286	0.320608	0.008985	0.000116
0.05	0.683078	0.315191	0.001723	0.000008
0.10	0.690079	0.309141	0.000780	0.000001
0.15	0.687986	0.311472	0.000541	0.000000

訂正処理をpass2までで終了とし，誤りが残っているかの検証にはBICONF[10]を用いる．BICONF[10]の処理手順を以下に示す．

1. pass2終了後にランダム関数を用いて系列の並びをスクランブルする．
2. 二項分布 $Bin(n,1/2)$ に基づき，系列を2つのブロックに分割する．
3. どちらか一方のブロックのパリティを求め，AliceとBobは公衆通信路を用いてパリティの比較を行う．
4. パリティが異なる場合には，双方のブロックにBINARYを施し，それぞれ1つの誤りを訂正する．

上述した処理を10回連続で誤りが検出されなくなるまで再帰的に行う．

最適化を施したブロックサイズを使用した場合の並列型Cascadeに対して，誤り率を0.0001から0.15まで0.0001ずつ変動させ，各誤り率に対して10,000回の計算機シミュレーションを行なった．Efficiencyの場合には0.01から0.15までを表示範囲とする．その結果をもとに，Efficiencyとround communicationについて考察する．

図3にEfficiencyを示す．式(4),(5)で与えられるブロックサイズを用いた並列型Cascadeと比較して，Opt.1，Opt.2ともに約10%程度Efficiencyが向上していることが確認できる．また，従来型CascadeのEfficiencyと比較してもより

Efficiencyが優れていることが確認できる．

図4にround communicationを示す．最適化を施した場合でも，従来型Cascadeと比較して大幅に少ない通信回数で訂正処理を行えていることが確認できる．また，誤り率に依存せずほぼ一定である．しかし，並列型Cascadeと比較して3倍程度round communicationが増加していることが確認できる．これは，pass2終了時にBICONF[10]を行なっているためである．また，pass1系列で訂正される誤りの個数を多く見積もると，pass2で使用するブロックサイズが大きくなるため，通信回数が増加する．

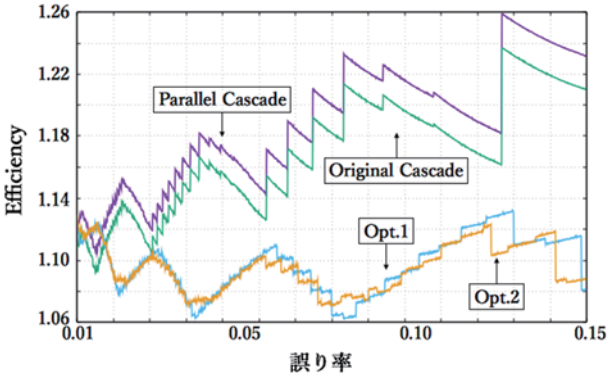

図3 Efficiency(従来型, 並列型, Opt.1, Opt.2)

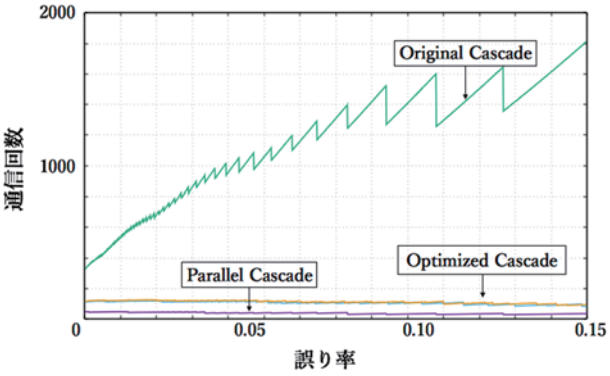

図4 通信回数(従来型, 並列型, Opt.1, Opt.2)

7. まとめ

並列型Cascadeの公開ビット数の削減を目的として，プロトコルに用いるブロックのサイズの最適化を行なった．その結果，公開ビット数を，10%程度削減することができたが，通信回数が並列型Cascadeの約3倍程度に増加した．

一方で，通信回数と公開ビット数のどちらを重視するかで，最適なブロックサイズの決定法が変化すると考えられる．

参考文献

1) C. H. Bennett and G. Brassard. ″Quantum cryptography: Public key distribution and coin tossing″. In Proceedings of IEEE International Conference on Computers, Systems and Signal Processing, volume 175, page 8. New York, 1984.

2) H.-K. Lo, "Method for decoupling error correction from privacy amplification", New J. Phys. 5 36, 2003.

3) Bennett, Charles H. et al. "Experimental quantum cryptography." Journal of Cryptology 5 3-28, 1990.

4) G. Brassard and L. Salvail, "Secret-key reconciliation by public discussion", Advances in Cryptology-Eurocrypt '93, edited by T. Helleseth, Lecture Notes in Computer Science, vol. 765 pp. 410-423, Springer, Berlin, 1994.

5) K. Yamazaki, R. Nair and H. P. Yuen, "Problems of the Cascade Protocol and its Application to Classical and Quantum Key Generation", Proc. of the 8th Int'l Conf. of Quantum Communication, Measurement and Computation, pp.201-204, NICT Press, 2006.

6) T. Odato and K. Yamazaki, "Properties of Secret-Key Reconciliation Protocol "Cascade" with Parallel Processing," NCSP16, pp.554-557, 2016.

2018年2月28日原稿受付， 2018年3月20日採録決定

Received, February 28, 2018; accepted, March 20, 2018