Cascadeプロトコルの通信回数の削減

A Study on Reduction of the Number of Communications for Cascade Protocol

福島拓*, 山﨑浩一*

Takumi Fukushima * and Koichi Yamazaki *

*玉川大学工学部ソフトウェアサイエンス科, 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

This paper proposes a modification scheme to reduce the number of public communications during the protocol for the interactive secret key reconciliation protocol ”Cascade”. The number of public communications would limit a key generation rate. In this modification, it is necessary to maintain the character of Cascade, which is the small amount of information exposed publicly as much as possible. The proposed protocol applies a binary search and block code together as error correction methods, taking the characters of the stages of passes which constitute the protocol into consideration. The amount of information exposed publicly and the number of public communications are given by computer simulation. The usefulness of the proposed protocol is demonstrated by comparing it with conventional schemes.

Keywords: secret key reconciliation, binary search, block error correcting code

1. はじめに

情報理論的に安全な暗号方式にバーナム暗号がある．これは，暗号通信を行う度に新しい暗号鍵を使用することを前提としている[1]．暗号鍵の事前共有の手法である量子暗号鍵配送は，量子通信路を介して送受信者が共有した二つの乱数系列に含まれる誤りの有無から盗聴を検出することで安全な鍵配送を実現する[1]．しかし，現実の量子通信路では盗聴者がいなくても量子通信路の雑音等によって誤りが発生する可能性がある．そのため，量子通信路で共有した乱数系列を秘密鍵として用いるためには，その誤りを訂正する必要がある．この誤りの訂正を秘密鍵系列の一致という[2]．秘密鍵系列の一致は公衆通信路を用いて行われる．この過程で送受信によって交換される乱数系列に関する情報を公開ビットという．公開ビット数の増加は最終的に得られる秘密鍵系列長の短縮につながるため，秘密鍵系列の一致では公開ビット数の少なさが求められる[2]．さらに，特に対話型の秘密鍵系列の一致では，通信回数の増加は鍵生成速度の低下の原因となるため，通信回数が少ないことも求められる[3]．秘密鍵系列の一致の代表的な対話型プロトコルにCascadeプロトコル[2]がある．Cascadeプロトコルは，公開ビット数が少ない長所がある一方で，誤り訂正に二分探索を用いるため通信回数が多い短所がある．そこで，Cascadeプロトコルの誤り訂正方式である二分探索の代わりにブロック符号を適用することが提案されている[4]（以降，“ブロック符号型プロトコル”）．ブロック符号型プロトコルでは，通信回数を約6割削減するが，公開ビット数は約1割増加してしまうことが示されている．

本研究では，Cascadeプロトコル(以降，“オリジナルプロトコル”)の通信回数を削減するとともに公開ビット数の増加を防ぐために，誤り訂正方式として二分探索とブロック符号を併用した対話型秘密鍵系列の一致プロトコルを提案し，計算機シミュレーションを用いて通信回数および公開ビット数について，オリジナルプロトコル，ブロック符号型プロトコルと比較し，性能を評価する．

2. 対話型秘密鍵系列の一致プロトコル

2-1 Cascadeプロトコル

Cascadeプロトコルは，Gilles BrassardとLouis Salvailによって提案された対話型の秘密鍵系列の一致プロトコルである[2)]．Cascadeプロトコルは以下に示すように複数のpassというステップを繰り返して乱数系列に含まれる誤りを削減していく．passの回数は実行前に送受信者間で決定する．本研究では文献2)と同様に，passの回数を4回とする．

いま，送受信者は，誤り率pの量子通信路を介して送られた長さNの二元乱数系列を共有しているものとする．

以下に，Cascadeプロトコルのアルゴリズムを記す．

Cascadeプロトコル

【pass 1】

1. 送受信者は，ブロックサイズk_1を定めて乱数系列をサイズk_1のブロックに分割する．
2. 送受信者は，公衆通信路を用いて各ブロックのパリティを比較する．
3. 送受信者は，パリティの異なる各ブロックに対して二分探索を用いて誤りを一つ訂正する．

【pass i $(i>1)$】

1. 送受信者は，乱数系列をランダムに並べ替えた後に，$k_i = 2k_{i-1}$のサイズのブロックに分割する．ただし，$2k_{i-1} > N/2$となる場合には$k_i = N/2$とする．
2. 送受信者は，公衆通信路を用いて各ブロックのパリティを比較し，パリティの異なるブロックを要素とする集合κを作成する．
3. 送受信者は，κに含まれるサイズが最も小さいブロック一つに対して，二分探索を用いて誤りを一つ訂正する．i以下の全てのpassの系列において，この訂正したビットを含む全てのブロックを要素とする集合βを作成する．
4. 送受信者は，集合$\kappa'=\beta \cup \kappa \setminus \beta \cap \kappa$とする．ここで，$\setminus$は差集合を表す．ここで，$\kappa' \neq \phi$であれば，$\kappa'$を新たな$\kappa$として手順3へ戻る．
5. $i \neq 4$であれば$i+1$を新たなiとして手順3へ戻る．$i = 4$であればプロトコルを終了する．

pass 1のブロックサイズk_1を以下の式を満たす最小の整数とする[2)]．

$$k_1 p - \frac{(1-(1-2p)^{k_1})}{2} \leq -\frac{\ln\frac{1}{2}}{2} \qquad (1)$$

ここで，ln·は自然対数である．

このように，Cascadeプロトコルでは，すでに終了したpassの系列ではすべてのブロックのパリティが一致していることを以降のpassで利用する連鎖的な誤り訂正により，公開ビット数を少なくすることに成功している．一方で，この連鎖的な誤り訂正を行うために，pass 2以降では，パリティの異なるブロックを一つずつ誤り訂正する必要があり，その結果，通信回数が多くなる問題が発生する．

2-2 ブロック符号型プロトコル

本節では，オリジナルプロトコルの通信回数を削減するために，オリジナルプロトコルの誤り訂正方式の二分探索をブロック符号に置き換えたプロトコルについて説明する[4)]．二分探索による誤り訂正では，サイズlのパリティの異なるブロックの誤りを訂正するために，送受信者は平均して約$\log_2 l$回の公衆通信路上の通信をする必要がある．一方，ブロック符号を用いた場合では，ブロックのサイズによらず必要な通信回数は1回である．多くの誤りを個別に訂正するCascadeプ

ロトコルでは，誤り訂正を二分探索からブロック符号にすることで全体の通信回数の削減が見込まれ，文献4)では，ブロック符号を用いたプロトコルを提案している．ここでは，単一の誤りを訂正するために必要な検査ビット数が最小の単一誤り訂正完全符号であるハミング符号を用いている[5]．ハミング符号では，符号長nと検査ビット数mの間に以下の関係がある．

$$n = 2^m - 1 \qquad (2)$$

このプロトコルでは，誤り訂正が行われるブロックには必ず誤りがある．この特徴を利用するためにパリティを比較するブロックのサイズk_1を，式(1)を満たす整数以下の$n+1=2^m$とする[6]．すなわち，ブロック符号型プロトコルでは，オリジナルプロトコル用のブロックサイズ以下で最大の2のべき乗の数をブロックサイズk_1とする．以下に，ブロック符号を用いてパリティの異なるブロックの誤りを訂正する方式の処理の流れを記す．

ブロック符号の適用方法 [6]

1. 送信者は，長さ$n-m$の二元乱数系列を生成し，情報ブロック$\boldsymbol{u}(\in\{0,1\}^{n-m})$とする．この情報ブロック$\boldsymbol{u}$に対してハミング符号化し，長さ$n$の符号語$\boldsymbol{w}$を生成する．ここで生成した二元乱数系列は，量子通信路で共有した乱数系列とは独立なものである．
2. 送信者は，パリティの異なる長さ$n+1$のブロック$\boldsymbol{A}$の最終ビットを取り除いた長さnの系列$\boldsymbol{A'}$を生成する．
3. 送信者は，送信語$\boldsymbol{x}=\boldsymbol{A'}\oplus\boldsymbol{w}$を生成し，公衆通信路を用いて受信者に送信する．ここで，"⊕"はビットごとの排他的論理和を表す．
4. 受信者は，対応するブロック$\boldsymbol{B}$の最終ビットを取り除いた$\boldsymbol{B'}$と送信者から送られてきた$\boldsymbol{w}$を用いて受信語$\boldsymbol{y}=\boldsymbol{B'}\oplus\boldsymbol{x}$を生成する．
5. 受信者は，受信後$\boldsymbol{y}$に対するシンドローム$\boldsymbol{s}$を求め，シンドローム$\boldsymbol{s}$より誤り位置を推定し，そのビットを反転する．ここで，$\boldsymbol{s}=\boldsymbol{0}$のときにはブロック$\boldsymbol{B}$の最終ビットを反転する．

上記の手順 5 において，$\boldsymbol{s}=\boldsymbol{0}$の場合には受信後$\boldsymbol{y}$には誤りがないものと推定することができる．したがって$\boldsymbol{B'}$には誤りがないものと考える．ここで，先述したように，誤り訂正が行われたブロック$\boldsymbol{B}$には必ず誤りが含まれているので，手順 4 で$\boldsymbol{B'}$を用意する際に$\boldsymbol{B}$から削除した最終ビットに誤りがあるものと推定する．

この誤り訂正の過程で盗聴者に漏れる情報量について考える[7]．簡単のために，盗聴者はブロック$\boldsymbol{B}$に関する情報を一切持っていないものと仮定する．このとき，長さ$n+1$のブロック$\boldsymbol{A}$に関する盗聴者のあいまいさ$H(\boldsymbol{A})$は次式で与えられる．

$$H(\boldsymbol{A}) = \log_2 2^{n+1} = n+1 \ \text{[bits]} \qquad (3)$$

盗聴者は公衆通信路を盗聴することで送信者から送られた送信語$\boldsymbol{x}=\boldsymbol{A'}\oplus\boldsymbol{w}$を知ることができる．この時，系列$\boldsymbol{w}$は，情報ブロック$\boldsymbol{u}$から一意に生成されるため，$\boldsymbol{w}$の候補数は$\boldsymbol{u}$の候補数，すなわち，$2^{n-m}$個となる．したがって，$\boldsymbol{x}$を知った盗聴者は系列$\boldsymbol{A'}$の候補を$2^{n-m}$個に絞り込むことができる．送信者が系列$\boldsymbol{A'}$を作るために$\boldsymbol{A}$から削除した最終ビットは等確率で発生する 2 値乱数である．その結果，最終的に$\boldsymbol{A}$の候補は等確率で発生する$2^{n-m}\times 2 = 2^{n-m+1}$個の系列となる．以上より，$\boldsymbol{x}$を知った盗聴者のブロック$\boldsymbol{A}$に関するあいまいさ$H(\boldsymbol{A}|\boldsymbol{x})$は次式となる．

$$H(\boldsymbol{A}|\boldsymbol{x}) = \log_2 2^{n-m+1} = n-m+1 \ \text{[bits]} \quad (4)$$

式(3)，(4)より，送信語$\boldsymbol{x}$を知ることで盗聴者に漏れる情報量$I(\boldsymbol{A};\boldsymbol{x})$は次式で与えられる．

$$\begin{aligned} I(\boldsymbol{A};\boldsymbol{x}) &= H(\boldsymbol{A}) - H(\boldsymbol{A}|\boldsymbol{x}) \\ &= (n+1) - (n-m+1) \\ &= m \qquad \text{[bits]} \end{aligned} \qquad (5)$$

ブロック符号を用いたプロトコルの問題点として，ブロックに複数個の誤りが含まれるときに誤りでないビットを反転する誤訂正が発生することが挙げられる．その結果，オリジナルプロトコルの二分探索を単にブロック符号に置き換えただけでは，誤訂正をしたブロックを再度訂正することによる無限ループに陥る可能性がある．以

下に，誤訂正の影響を考慮することで無限ループが発生しないプロトコルを示す．

ブロック符号型プロトコル[4)]

【pass 1】

1. 送受信者は，ブロックサイズk_1を定めて乱数系列をサイズk_1のブロックに分割する．
2. 送受信者は，公衆通信路を用いて各ブロックのパリティを比較する．
3. 送受信者は，パリティの異なる各ブロックに対して，ブロック符号を用いて誤りと推定されるビットを求め，そのビットを反転する．

【pass i $(i>1)$】

1. 送受信者は，乱数系列をランダムに並べ替えた後に，$k_i = 2k_{i-1}$のサイズのブロックに分割する．ただし，$2k_{i-1} > N/2$となる場合，$k_i = N/2$とする．
2. 送受信者は，公衆通信路を用いて各ブロックのパリティを比較し，パリティの異なるブロックを要素とする集合κを作成する．さらに，空集合μを作成する．
3. 受信者は，$\kappa \setminus \mu$内のブロックサイズが最も小さいブロックを一つ選択する．ここで，$\kappa \setminus \mu = \emptyset$であり，選択できるブロックがないとき，手順6へ進む．
4. 送受信者は，手順3で選択したブロックをμに入れ，ブロック符号を用いて誤りと推定されるビットを求め，そのビットを反転させる．i以下のすべてのpassの系列におけるこの反転したビットを含むブロックを要素とする集合βを作成する．
5. 受信者は，集合$\kappa' = (\beta \cup \kappa) \setminus (\beta \cap \kappa)$とする．$\kappa' = \emptyset$のとき，pass iを終了する．$\kappa' \neq \emptyset$の場合，κ'をκに置き換えて手順3へ戻る．
6. $i \neq 4$であれば$i+1$を新たなiとして手順3へ戻る．　$i = 4$であればプロトコルを終了する．

3. 提案型プロトコル

ブロック符号型プロトコルでは，オリジナルプロトコルに比べて通信回数が削減された一方で，公開ビット数は増加した．その原因はパリティの異なるブロックに複数個の誤りがあるときに発生する誤訂正である．また，オリジナルプロトコル，ブロック符号型プロトコルともに，pass 1の通信回数が全通信回数に占める割合は非常に少ない．これは，pass 1ではパリティの異なるすべてのブロックの誤り訂正を同時に行うことができるが，pass 2以降では，連鎖的な誤り訂正を行うために1ブロックずつ誤りを訂正する必要があるためである．そこで，本稿では全体に占める通信回数の割合が限定的であるpass 1において，通信回数は増えるが誤訂正を起こさない二分探索法を用い，pass 2以降では，通信回数の少ないブロック符号を併用するプロトコルを提案する．プロトコルの処理の流れはブロック符号型プロトコルのpass 1をオリジナルのものに置き換えるだけであるので，提案型プロトコルの処理の流れをここに改めて記すことは控える．

4. 計算機シミュレーション結果と考察

オリジナルプロトコルとブロック符号型プロトコル，提案型プロトコルについて，量子通信路の誤り率pを0.01から0.15まで0.01ずつ変えたときに計算機シミュレーションよって求められた公開ビット数を図1に示す．系列の長さNを10,000とし，各プロットは10万回の試行結果の平均値である．

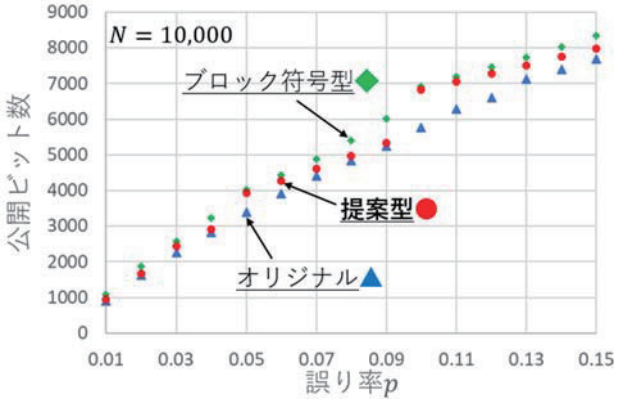

図1 公開ビット数

図1より，提案型プロトコルはブロック符号型プロトコルと比較して，全ての誤り率で公開ビット数が削減していることがわかる．ある誤り率で提案型の公開ビット数がオリジナルプロトコルに接近し，そこから誤り率が増加するとその差が急激に開き，その後，誤り率の増加に伴いオリジナルプロトコルの値に近づくことが繰り返されている．その原因を明らかにするために，本稿で扱っているプロトコルにおける公開ビットの発生について考察する．ここで扱っているプロトコルでは，公開ビットは，各パスの最初に行う各ブロックのパリティの比較とその後に行われるパリティの異なるブロックの誤り訂正の二つの処理において発生する．これらの二つの処理において発生する公開ビットの数はともにブロックサイズに依存する．そこで，誤り率の変化に伴うブロックサイズの推移に着目する．表1に計算機シミュレーションにおいて3つのプロトコルで使用したブロックサイズを示す．表1より，図1において提案型プロトコルの公開ビット数がオリジナルプロトコルに接近した誤り率p =0.04，0.09において，提案型とオリジナルプロトコルのブロックサイズが同じか近い値であることがわかる．それぞれの点において誤り率が0.01増えるとオリジナルプロトコルではブロックサイズがわずかに減少しているのに対して提案型では半分の値になっている．誤り率をより細かく増やした場合には，オリジナルプロトコルではブロックサイズは1ずつ減少するのに対して，提案型ではこの場合でも半減する．これは，2-2節で述べたように提案型ではハミング符号を使用しており，その結果，ブロックサイズが2のべき乗に制限されるためである．ここで，ブロックパリティの比較のための公開ビット数はそのpassのブロック数$\lceil N/k_i \rceil$であるため，ブロックサイズが半減するとその数は2倍になる．その結果，提案型プロトコルでは公開ビット数の段階的な増加が発生する．

表 1 オリジナル，ブロック符号型，提案型 Cascade プロトコルのブロックサイズk_i

誤り率p	ブロックサイズk_1	
	オリジナル	ブロック符号型提案型
0.01	73	64
0.02	36	32
0.03	24	16
0.04	18	16
0.05	14	8
0.06	12	8
0.07	10	8
0.08	9	8
0.09	8	8
0.1	7	4
0.11	6	4
0.12	6	4
0.13	5	4
0.14	5	4
0.15	5	4

3 つのプロトコルの通信回数について図 1 と同じ条件で行った計算機シミュレーション結果を図 2 に示す．図 2 より，提案型プロトコルでは，ブロック符号型プロトコルと比較して，全ての誤り率で通信回数が削減していることがわかる．本稿で行った計算機シミュレーションの誤り率では，オリジナルプロトコルと比較して提案型プロトコルは通信回数を少なくとも約 8 割削減している．提案型プロトコルは pass 1 で二分探索法を用いているため，pass 1 だけを考えるとブロック符号型プロトコルより通信回数は多い．しかし，提案型では pass 1 で二分探索を用いたことで pass 1 において誤訂正が発生しない．その結果，

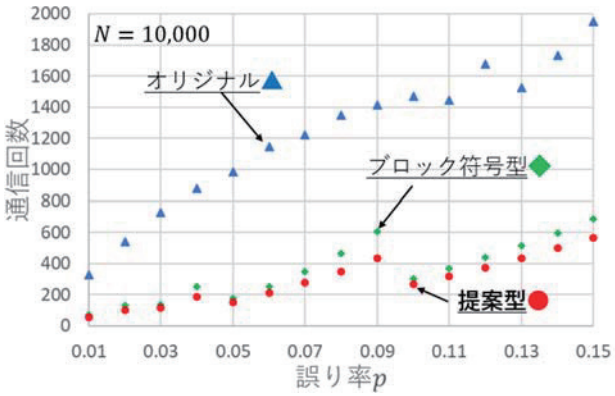

図2 通信回数

表 2 計算機シミュレーションにおいて各パスで発生した誤訂正数

(a) ブロック符号型プロトコル

誤り率 p	誤訂正数				
	pass 1	pass 2	pass 3	pass 4	総数
0.01	3.588	5.517	2.267	0.025	11.397
0.02	6.999	10.581	3.685	0.004	21.270
0.03	6.404	7.952	0.759	0.000	15.114
0.04	13.350	19.660	5.786	0.001	38.796
0.05	6.786	7.690	0.224	0.000	14.700
0.06	11.120	13.351	0.832	0.000	25.303
0.07	16.802	21.683	2.724	0.000	41.209
0.08	23.836	33.528	7.606	0.000	64.970
0.09	32.168	49.697	18.174	0.001	100.040
0.1	8.993	9.725	0.118	0.000	18.837
0.11	11.843	13.054	0.234	0.000	25.131
0.12	15.200	17.137	0.464	0.000	32.801
0.13	19.121	22.144	0.906	0.000	42.170
0.14	23.615	28.193	1.699	0.000	53.507
0.15	28.693	35.450	3.089	0.000	67.233

(b) 提案型プロトコル

誤り率 p	誤訂正数				
	pass 1	pass 2	pass 3	pass 4	総数
0.01	—	1.374	0.149	0.000	1.523
0.02	—	2.496	0.151	0.000	2.648
0.03	—	1.199	0.010	0.000	1.210
0.04	—	4.344	0.145	0.000	4.488
0.05	—	0.759	0.002	0.000	0.761
0.06	—	1.732	0.007	0.000	1.739
0.07	—	3.503	0.027	0.000	3.530
0.08	—	6.549	0.111	0.000	6.660
0.09	—	11.436	0.428	0.000	11.865
0.1	—	0.592	0.002	0.000	0.593
0.11	—	0.946	0.002	0.000	0.948
0.12	—	1.467	0.004	0.000	1.471
0.13	—	2.198	0.007	0.000	2.205
0.14	—	3.211	0.012	0.000	3.223
0.15	—	4.580	0.025	0.000	4.605

pass 2 以降で訂正される誤りの個数が減少したことが通信回数を押し下げた要因と考えられる．そこで，ブロック符号型と提案型プロトコルの計算機シミュレーションにおいて各 pass で発生した誤訂正数を表 2 に示す．

表2より，提案型プロトコルではpass 1に二分探索を用いることでそこでは誤訂正が発生しないために, pass 2以降で発生する誤訂正数が減り，プロトコル全体における誤訂正の大幅な削減につながっていることがわかる．二つのプロトコルの誤訂正の総数を比較すると，今回の計算機シミュレーションで最も削減の割合が少ない$p = 0.01$の場合でも誤訂正は約86%削減している．連鎖的に誤りを訂正するプロトコルではpass 2以降で訂正する誤り数が通信回数に直結する．提案型プロトコルでは，pass 2以降に発生する誤訂正を大幅に減らし，pass 2以降で訂正する誤り数が削減されたことに伴い，ブロック符号型プロトコルより通信回数が減少したことがわかる．

6. まとめ

本研究では，対話型秘密鍵系列の一致であるCascadeプロトコルの短所である通信回数が多い点に着目し，公開ビット数の増加を抑えつつ，通信回数を減らすプロトコルを提案した．提案プロトコルは，Cascadeプロトコルにける各passの特徴を考慮して，誤り訂正方式に二分探索とブロック符号を併用するものである．計算機シミュレーションの結果，オリジナルのCascadeプロトコルより通信回数を大幅に削減することとともに，通信回数を削減するために提案されたブロック符号用いる既存のプロトコルと比較して，公開ビット数，通信回数をともに削減できることを示した．

参考文献

1) 岡本龍明， 山本博資: 現代暗号， 産業図書株式会社, (1997).

2) G. Brassard and L. Salvail: Advances in Cryptology-Eurocrypt ’93, edited by T. Helleseth, Lecture Notes in Computer Science, vol. 765, pp. 410-423, Springer, Berlin, (1994).

3) P. M. Nielsen, C. Schori, J. L. Sorensen, L. Salvail, I. Damgard, and E. Polzik: J. Mod. Opt. 48, pp.1921-1942 (2001).

4) 小宮山由布輝: 令和四年度玉川大学工学部ソフトウェアサイエンス学科卒業論文.

5) 和田山正: 誤り訂正技術の基礎, 森北出版, (2010).

6) E. Furukawa and K. Yamazaki: 2001 International Symposium on Communication and Information Technology (ISCIT 2001), Chiang Mai, Thailand,

pp.14-16, (2001).
7) 植松友彦: イラストで学ぶ情報理論の考え方, 講談社, (2012).

2024年3月11日原稿受付，2024年3月15日採録決定
Received, March 11th, 2024; accepted, March 15th, 2024