Block size 와 Pass 횟수에 따른 CASCADE 의 Failure rate 분석

김건후, 허준*
고려대학교, *고려대학교

giraffeaffe@korea.ac.kr, *junheo@korea.ac.kr

Analysis of Failure Rates with Different Block Sizes and Different Number of Passes

Kim Kun Hoo, Heo Jun*
Korea Univ., *Korea Univ.

요 약

본 논문은 BB84 프로토콜과 같은 Quantum Key Distribution 프로토콜을 이용하여 동일한 key 를 공유할 수 있도록 오류 정정을 수행하는 CASCADE 알고리즘에서 block size 와 전체 pass 횟수를 다르게 적용하였을 때 발생하는 차이를 시뮬레이션을 이용하여 비교 분석하였다.

Ⅰ. 서 론

CASCADE 알고리즘은 QKD 프로토콜 중 하나인 BB84 프로토콜에서 사용되는 알고리즘으로, Alice 가 Bob 에게 Quantum channel 을 통하여 전송한 양자 키에서 발생한 비트 오류를 Classical channel 을 이용하여 정정함으로써 서로 동일한 키를 공유할 수 있도록 하는 후처리 과정에서 사용된다.

본 논문에서는 CASCADE 알고리즘의 두 가지 파라미터인 block size 와 전체 pass 의 횟수에 따라 CASCADE 의 오류 정정 실패율이 어떻게 달라지는지를 [1], [2], [3]에서 각각 제시된 파라미터를 이용하여 비교한다.

Ⅱ. 본론

1. CASCADE 알고리즘 동작 원리

CASCADE 알고리즘은 QKD 프로토콜의 후처리 과정인 Information Reconciliation 과정에서 사용된다. Information Reconciliation 과정은 QKD 프로토콜을 사용하는 Alice 와 Bob 이 Quantum channel 을 통해 주고받은 key 에서 발생한 비트 에러를 정정하여 Alice 와 Bob 이 동일한 key 를 공유하도록 하여 QKD 프로토콜의 목표를 달성하기 위해 반드시 필요한 과정이다. 이때 오류 정정을 위해서는 Classical channel 을 사용하게 되며, Classical channel 은 Quantum channel 과 달리 도청자의 존재를 직접적으로 알 수 없기 때문에 Classical channel 을 사용한 통신으로 key 에 대한 parity 등의 정보를 주고받을 때 Information leakage 가 발생한다고 간주하게 된다.

CASCADE 의 기본적인 동작 구조는 다음과 같다.

(1) 주어진 key 를 일정한 크기로 나누어 block 들을 생성한다. (보통 두 번째 pass 부터는 block 으로 나누기 전에 key 에 random shuffling 을 수행함)
(2) 현재 pass 에서 새로 생성한 block 들을 모두 집합 S 에 저장한다.
(3) 새로 생성한 모든 block 들에 대해 Alice 와 Bob 의 parity 를 비교하는 parity check 와 오류 정정을 수행하고 해당 블록들을 S 에서 제거한다.
(4) S 에 포함된 이전 pass 의 블록들 중 앞선 오류 정정으로 달라진 bit 를 포함하는 블록들을 S 에서 제거하고, S 에 포함되어 있지 않던 블록 중에서 오류 정정으로 달라진 bit 를 가진 블록들을 S 에 포함시킨다. 따라서 S 는 홀수 개의 오류를 가진 블록들만 남게 된다.
(5) S 에 포함된 블록 중 가장 크기가 작은 블록에 대한 오류 정정을 수행하고 (4)의 과정을 다시 수행한다.
(6) S 에 남은 블록이 없을 때까지 (4)~(6)의 과정을 반복적으로 수행한다.

여기서 pass 란 (1)~(6)의 전체 과정을 말하며, CASCADE 알고리즘은 기본적으로 이러한 pass 를 여러 차례 수행함으로써 key 에 존재하는 모든 비트 에러를 제거할 수 있다. 또한, 위의 과정 중 (3)과 (5)의 과정에서의 오류 정정은 BINARY 알고리즘을 사용하여 진행된다. BINARY 알고리즘은 Binary search 와 유사하게 주어진 block 을 절반인 b_L과 b_R로 나누고, b_L에 대하여 Alice 와 Bob 의 parity 를 비교하여 만약 두 parity 가 동일하다면 block 의 나머지 절반인 b_R을 다시 반으로 나누어 parity 를 비교하는 동일한 과정을 반복하고, 만약 두 parity 가 다르다면 해당되는 b_L의 절반을 다시 반으로 나누고 parity 를 비교하는 과정을 반복하는 방식으로 최종적으로 주어진 block 에서 에러가 존재하는 위치 1 개를 얻는 방식으로 이루어진다. 따라서 CASCADE 알고리즘에서 각각의 block 에 대한 오류 정정은 단일한 bit 에 대하여 이루어지며, 오류 정정 과정에서도 parity check 가 이루어지므로 역시 Information leakage 가 발생한다.

2. CASCADE 알고리즘의 파라미터

Brassard 와 Salvail 의 논문 [1]에서 처음 제안된 CASCADE 알고리즘은 다음과 같은 block size 와 전체 pass 횟수를 사용한다.

$$k_1 = 0.73/\epsilon$$
$$k_2 = 2k_1$$
$$k_3 = 2k_2$$
$$k_4 = 2k_3$$

또한, [1] 이후에 보다 optimal 한 오류 정정을 위해 변형된 CASCADE 알고리즘들이 제안되었으며, [2]와 [3]의 경우 변형된 알고리즘과 함께 새로운 block size 와 전체 pass 횟수를 제시하였다. H. Yan 등이 [2]에서 제시한 block size 와 pass 횟수는 다음과 같으며,

$$k_1 = 0.8/\epsilon$$
$$k_2 = 5k_1$$
$$k_i = N/2 \ \ (i = 3,4,\cdots,10)$$

마지막으로 Martinez-Mateo 등이 [3]에서 제시한 block size 와 전체 pass 횟수는 다음과 같다.

$$k_1 = 2^{\lceil log_2\left(\frac{1}{Q}-\frac{1}{2}\right)\rceil}$$
$$k_2 = 2^{\lceil \left(log_2\left(\frac{1}{Q}-\frac{1}{2}\right)+12\right)/2\rceil}$$
$$k_3 = 4096$$
$$k_i = \lceil N/2 \rceil \ \ (i = 4,5,\cdots,14)$$

여기서 Q 와 ϵ은 모두 QBER(Quantum Bit Error Rate)를 나타낸다.

3. Failure rate 비교

본 논문에서는 Alice 와 Bob 이 공유하는 전체 key 에 대한 오류 정정 실패율을 확인하기 위하여 C 를 이용한 시뮬레이션을 사용하였다. 시뮬레이션에서는 QBER 을 0.001 부터 0.1 까지 0.001 씩 증가시키며, 각 QBER 마다 동일한 파라미터로 10000 bit 길이의 key 에 대한 오류 정정을 100 회씩 진행하여 전체 key 에 대한 오류 정정이 실패한 경우를 카운트하는 방식으로 진행하였다.

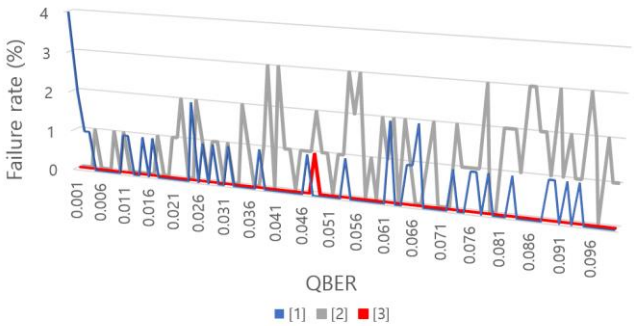

그림 1

그림 1 은 [1], [2], [3]에서 각각 제시한 block size 와 전체 pass 횟수를 적용하였을 때 얻은 결과를 보여준다. 여기서 사용된 CASCADE 알고리즘은 [2]와 [3]에서 제안한 변형된 알고리즘을 사용하지는 않고, 동일한 조건에서의 비교를 위해 [1]에서 제안된 가장 기본적인 구조를 따랐다.

CASCADE 알고리즘을 처음 제안한 [1]의 파라미터를 사용한 경우, QBER 이 0 에 가까우면 실패율이 커지는 경향을 확인할 수 있다. 이는 [1]에서는 초기 block size 를 $k_1 = 0.73/\epsilon$ 로 결정하고 이후 pass 에서는 block size 를 두 배로 증가시키는 방식으로 block size 를 결정하기 때문에 발생하는 문제이다. CASCADE 알고리즘에서 오류 정정을 위해 사용하는 BINARY 알고리즘은 각 block 마다 1 개의 오류만을 정정할 수 있기 때문에, 4 회의 pass 만을 사용하는 [1]의 조건에서는 QBER 이 0 에 가까울 수록 block 의 크기가 지나치게 커져 한 번의 pass 에서 정정할 수 있는 오류의 수가 줄어들며, 큰 block 을 사용하면 오류가 발생한 bit 가 random shuffling 을 할 때마다 동일한 block 에 위치하게 될 확률이 높아지기 때문에 정밀한 오류 검출을 위해서는 더 많은 양의 pass 가 필요하게 된다. 따라서 처음부터 [1]보다 큰 크기의 block 들을 사용하도록 고안된 [2]와 [3]의 경우 각각 10 회, 14 회라는 많은 횟수의 pass 를 진행하도록 하였기 때문에 QBER 이 0 에 가까우면 실패율이 증가하는 경향을 보이지는 않는 것을 확인할 수 있다. 특히, [3]에서 제시된 block size 와 전체 pass 횟수를 적용했을 때 모든 QBER 에 대해서 낮고 안정적인 실패율을 보이는 것을 확인할 수 있다.

Ⅲ. 결론

본 논문에서는 CASCADE 알고리즘을 처음 제안한 논문 [1]에서 제시된 block size 와 pass 횟수를 사용하지 않고, [2]와 [3]에서 제시된 방식으로 [1]보다 큰 block 과 더 많은 pass 를 사용하는 경우, QBER 이 0 에 가까울 때 [1]보다 낮은 failure rate 를 얻을 수 있음을 확인하였으며, 특히 [3]에서 제시된 block size 와 pass 횟수를 사용할 경우 다른 경우보다 failure rate 의 관점에서 이득을 얻을 수 있음을 확인하였다.

ACKNOWLEDGMENT

본 연구 논문은 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임 (No. 2020-0-00014, 결함허용 논리양자큐빗 환경을 제공하는 양자운영체제 원천기술 개발)

본 논문은 2021 년도 ㈜ 케이티의 지원을 받아 수행되었음.

참 고 문 헌

[1] Brassard G., Salvail L. (1994) Secret-Key Reconciliation by Public Discussion. In: Helleseth T. (eds) Advances in Cryptology — EUROCRYPT '93. EUROCRYPT 1993. Lecture Notes in Computer Science, vol 765. Springer, Berlin, Heidelberg. https://doi.org/10.1007/3-540-48285-7_35

[2] H. Yan et al., "Information Reconciliation Protocol in Quantum Key Distribution System," 2008 Fourth International Conference on Natural Computation, Jinan, China, 2008, pp. 637-641, doi: 10.1109/ICNC.2008.755.

[3] Jesus Martinez-Mateo, Christoph Pacher, Momtchil Peev, Alex Ciurana, and Vicente Martin. 2015. Demystifying the information reconciliation protocol cascade. Quantum Info. Comput. 15, 5- 6 (April 2015), 453- 477.