Toward an Efficient Website Fingerprinting Defense 논문 리뷰

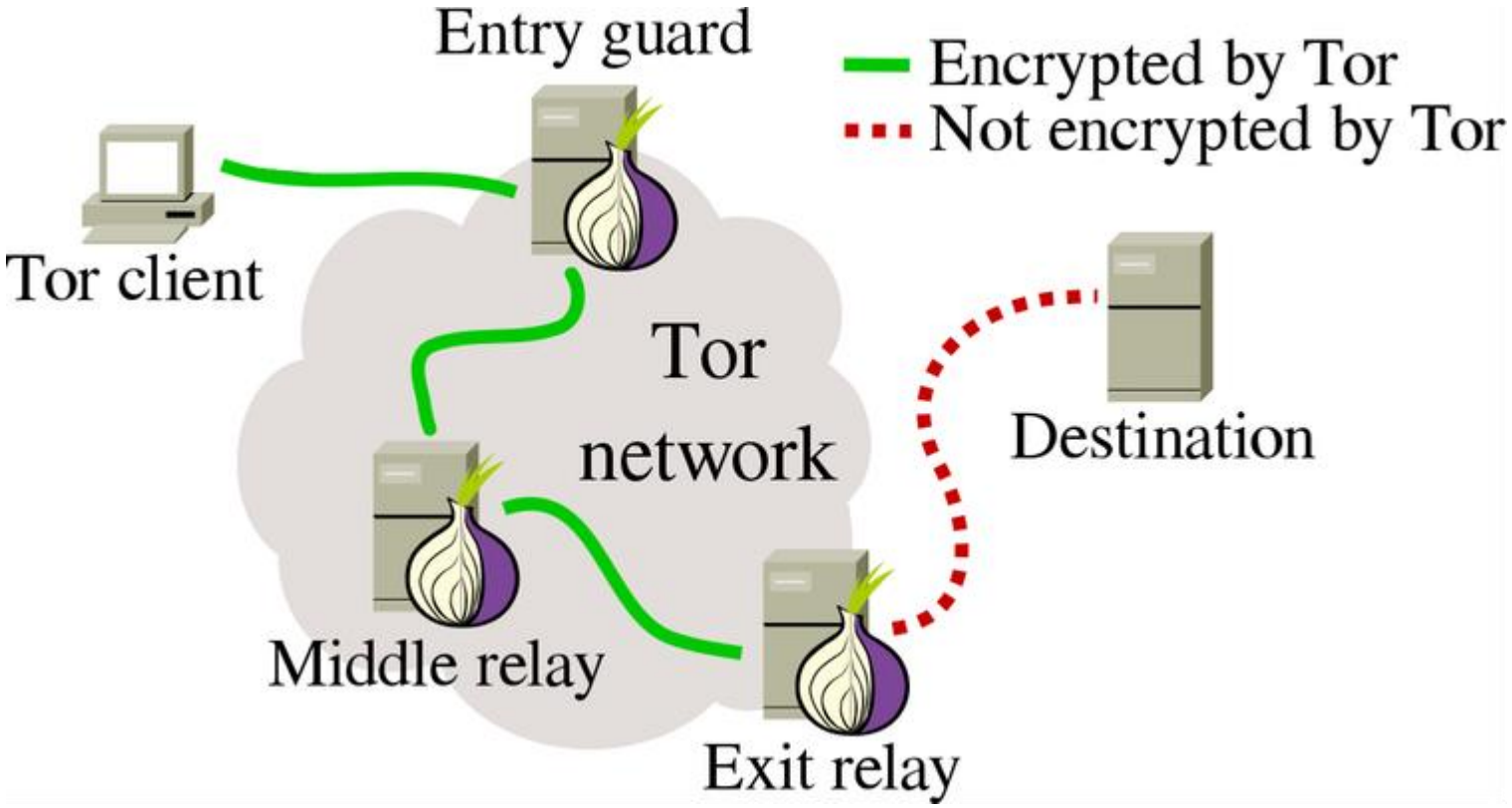
https://youtu.be/57ESdHVBbps

HSU 한성대학교 HANSUNG UNIVERSITY

CryptoCraft LAB
https://crypto.modoo.at

Tor(*The Onion Router*) network

- 인터넷 사용자에게 **익명성**을 제공하는 데
  가장 많이 사용되는 인기있는 시스템



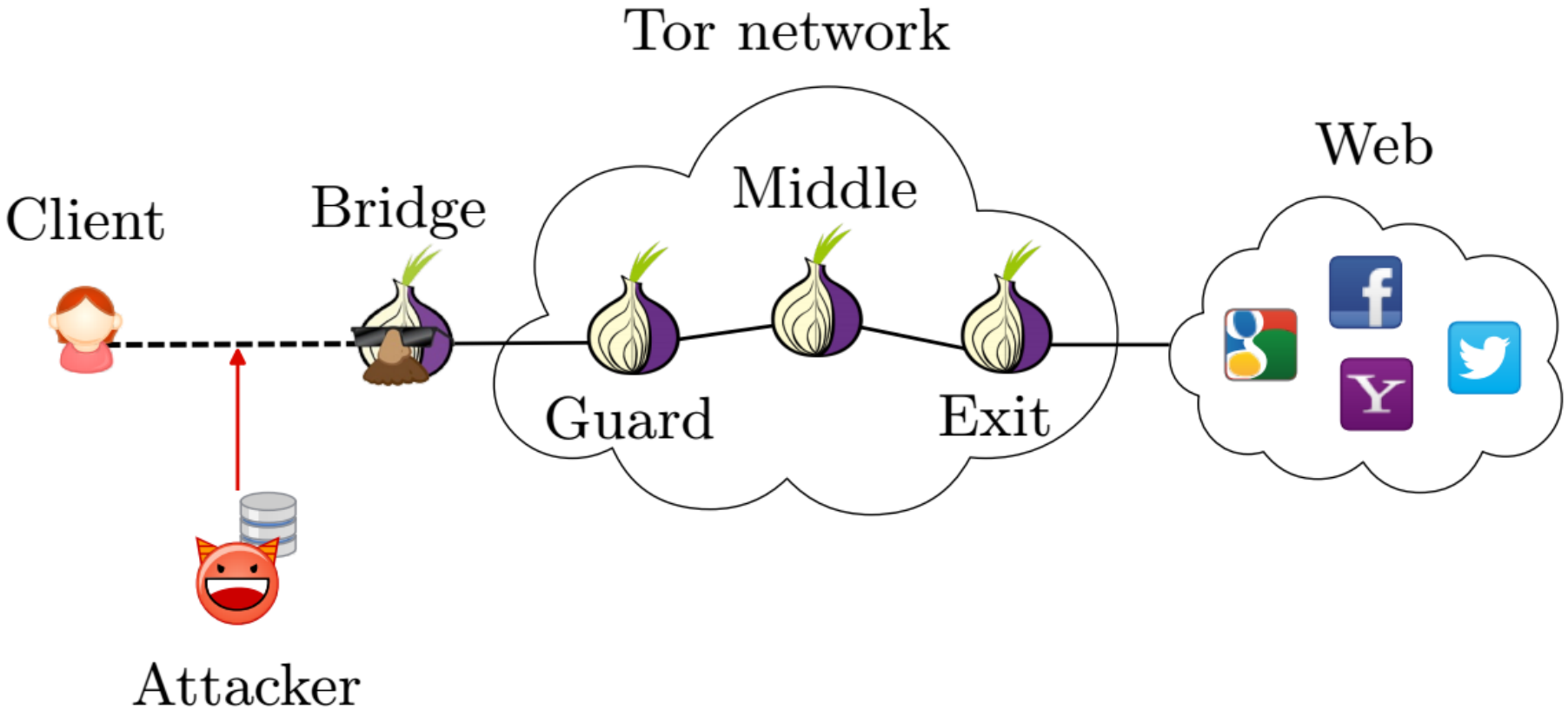
Tor(*The Onion Router*) network

- Tor는 목적지까지 한 번에 통신하지 않고,
  중간에 같은 tor 라우터를 실행하고 있는
  node들을 여러 개 거쳐서 보냄

- 패킷을 양파 껍질처럼 **겹겹이 암호화** 해서 보내고,
  이때 **각각의 node의 공개키를 통해 암호화**하므로,
  패킷의 출발지와 최종 목적지를 알아내려면
  거의 모든 노드를 장악해야 한다.

Web Fingerprinting attack

- 공격자가 관찰된 트래픽과 사전에 기록된 웹 트래픽 템플릿을 통해 웹 브라우징 활동을 복구하는 공격



Web Fingerprinting attack

- Tor에 대한 첫 번째 공격은 폐쇄 된 세계에서 WF 방어 없이 Naive Bayes 분류기로 3 %의 정확도
- 최신 공격은 90 % 이상의 정확도를 달성
- Tor에 사용하기 위한 WF 방어 필요

WF Defenses

- 어플리케이션 레벨 방어

  -HTTPOS
  응용 계층에서 HTTP 헤더를 수정하고 HTTP 요청을 전략적으로 주입
  -tor에 구현된 Randomized Pipelining
  HTTP 요청의 파이프 라인을 무작위 화
  -여러 평가에서 효과 X

- 슈퍼 시퀀스(최적의 공통 시퀀스) 및 트래픽 모핑

  - 트래픽을 다른 클래스처럼 보이도록 최적으로 변형
  - 웹 페이지 템플릿 데이터베이스를 필요 함
  자주 업데이트되어 유지 관리 비용이 많이 듦

WF Defenses

- 일정 속도로 패딩 방어

  BuFLO : 고정 크기 패킷으로 일정한 속도의 트래픽을 사용
  -방어 시간보다 2 ~ 3 배 더 긴 대기 시간으로
  오버 헤드가 매우 높으며 대역폭 오버 헤드는 100 % 이상
  -동적 웹 컨텐츠의 인기로 인해 페이지로드가
  완료되는 시점을 결정하기가 어렵움

WF Defenses

- Adaptive Padding

- AP는 방어자가 나가는 트래픽 패턴을 검사하고 패턴의 특징을 방해하기 위해 목표 방식으로 **더미 메시지를 생성**
- BuFLO기반의 방어에서는 패킷 간 간격 시간이 고정되어 일정한 패킷 타이밍 일정에 맞게 **애플리케이션 데이터가 지연됨** Tor와 같은 시스템에 적합하지 않음
- 적응 형 패딩 (AP)은 응용 프로그램 데이터를 **지연시키지 않음** Tor의 훌륭한 후보

Adaptive Padding

- 실제 패킷 흐름은 **폭주하는 경향**이 있으며,
  각 흐름은 자연스럽게 패킷 간 간격의 **고유 한 패턴**을 가지므로
  공격자가 "지문"으로 사용할 수 있음

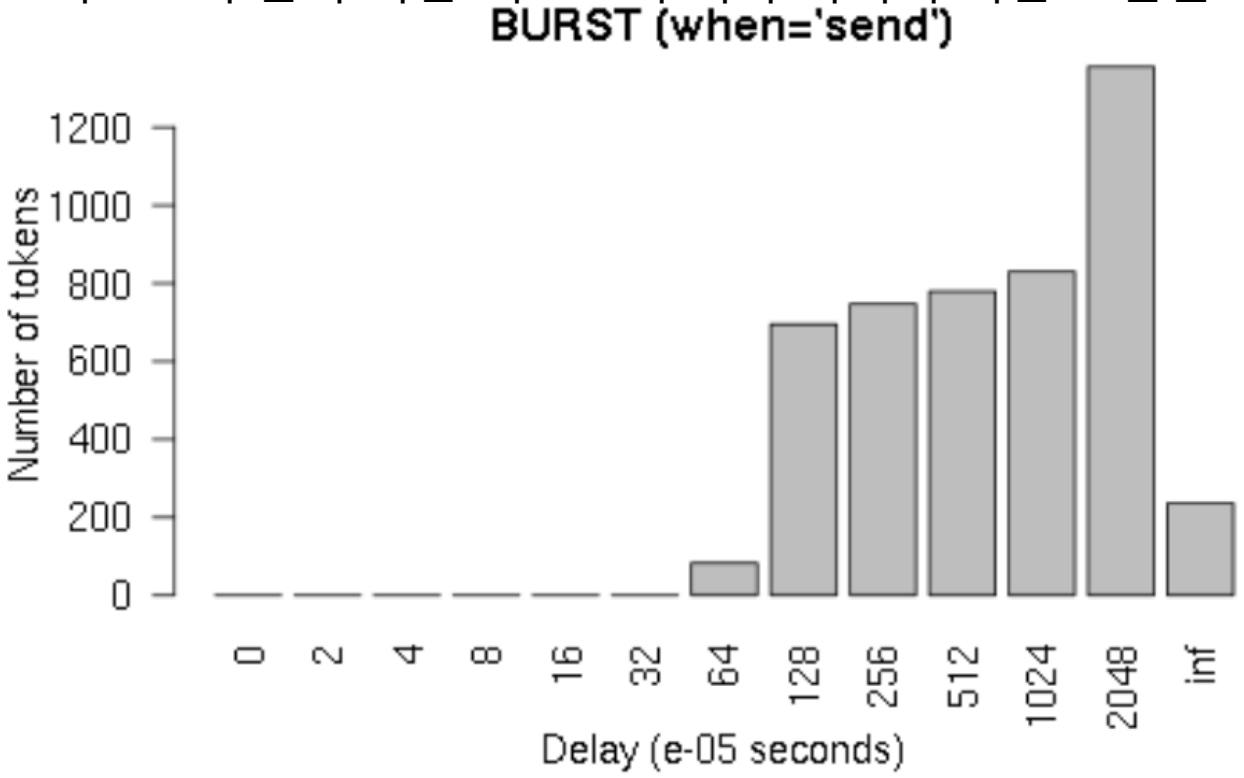
- 자연스러운 지문을 파괴 하기위해서 비정상적으로
  큰 간격이 발견되면 **해당 간격에 패딩을 추가**하여
  긴 간격이 구별되는 특징이 되는 것을 방지

- 이중 모드 알고리즘
  burst 모드
  gap 모드

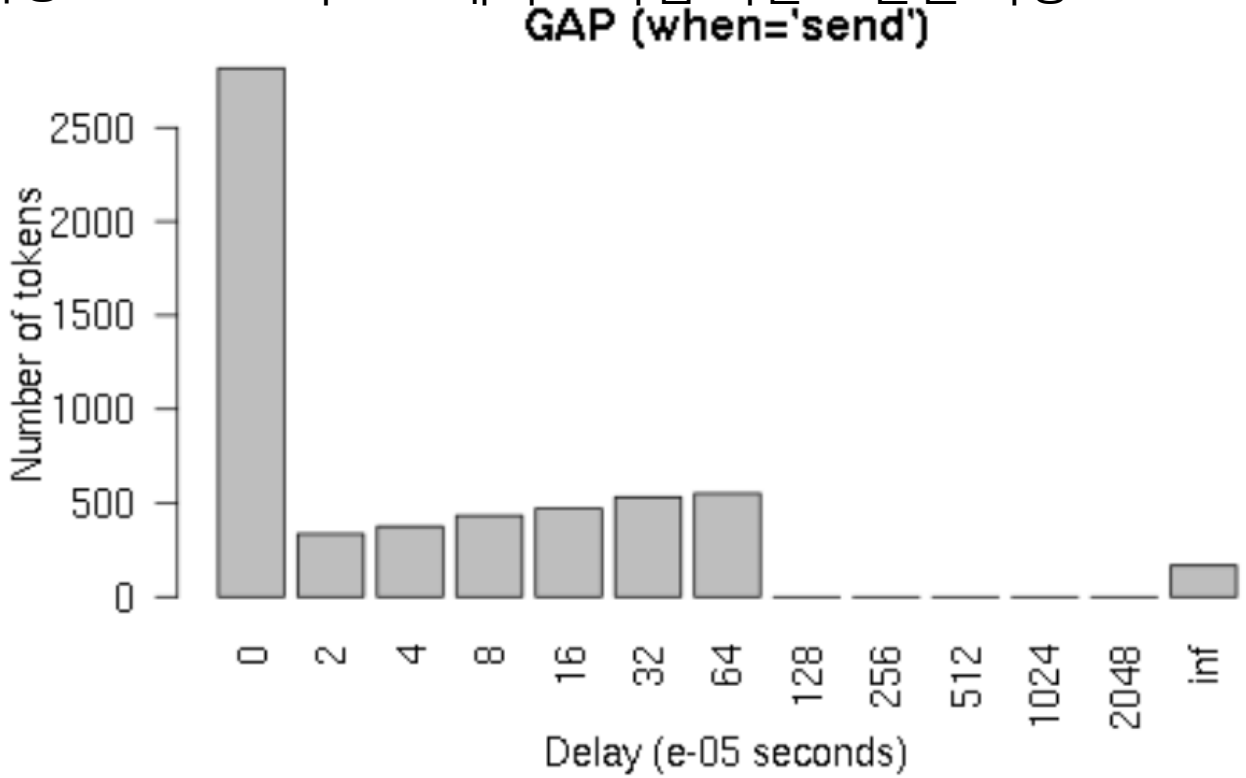
Adaptive Padding

- 두 개의 지연 히스토그램
- "정상"흐름에 대한 대략적인 통계 분포를 미리 수집

버스트의 끝과 다음 버스트의 시작 사이의 시간 표본을 사용

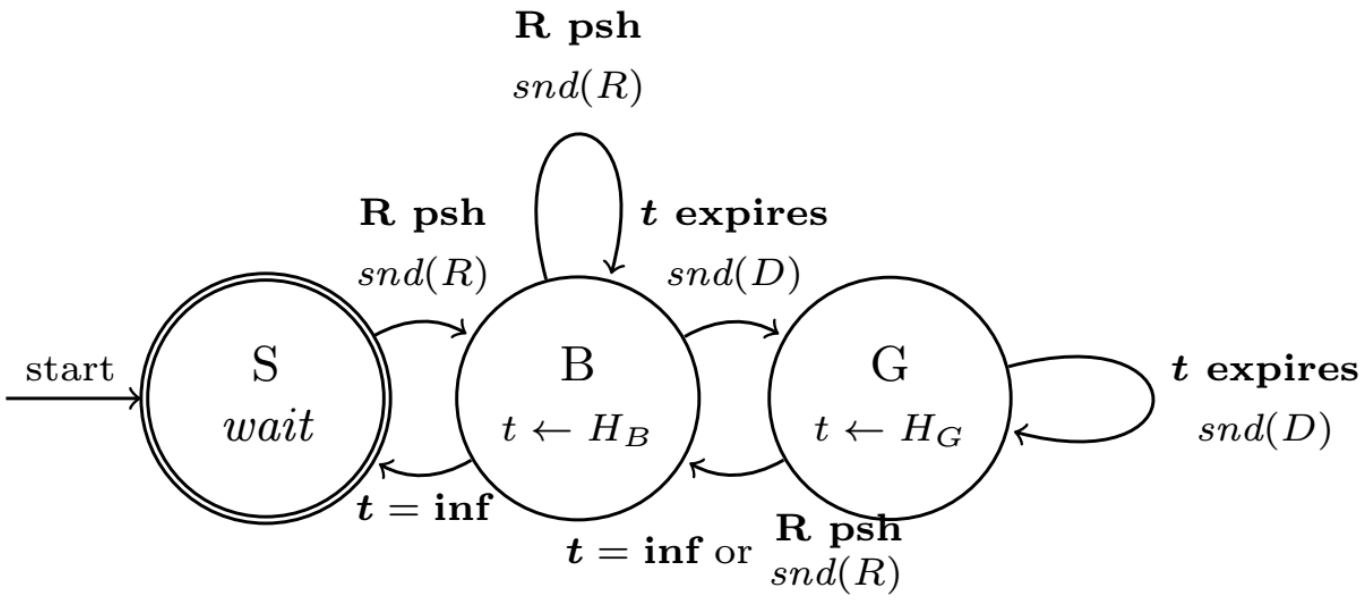


버스트 내의 교차점 시간 표본을 사용



Adaptive Padding

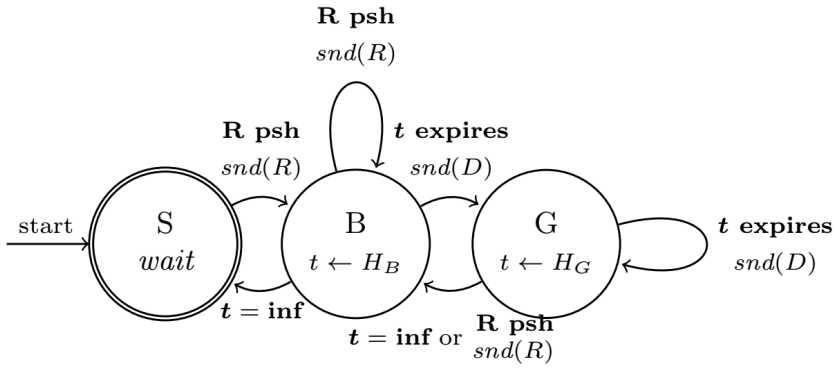
```
def update_state(self, packet, flow):
    """Switch state accordingly to AP machine state."""
    if flow.state == WAIT and not packet.dummy:
        flow.state = BURST

    elif flow.state == BURST and flow.expired:
        flow.state = GAP

    elif flow.state == BURST and flow.timeout == INF:
        flow.state = WAIT

    elif flow.state == GAP and flow.timeout == INF:
        flow.state = BURST

    elif flow.state == GAP and not packet.dummy:
        if self.stop_on_real:
            flow.state = WAIT

    else:
        return False
```



Adaptive Padding

```python
def add_padding(self, i, trace, flow, on):
    """Generate a dummy packet."""
    packet = trace[i]

    if flow.state == WAIT:
        return

    timeout = INF
    histogram = self.hist[flow.state][flow.direction][on]
    if histogram is not None:
        timeout = histogram.random_sample()

    try:
        iat = self.get_iat(i, trace, flow.direction, on)
    except IndexError:
        self.pad_end_flow(flow)
        return
```

```python
    # if iat <= 0 we do not have space for a dummy
    if not iat <= 0:
        if timeout < iat:
            logger.debug("timeout = %s  < %s = iat", timeout, iat)

            # timeout has expired
            flow.expired, flow.timeout = True, timeout

            # the timeout has expired, we send a dummy packet
            dummy = self.generate_dummy(packet, flow, timeout)

            # correct the timeout
            iat = timeout

            # add dummy to trace
            insort_left(trace, dummy)

    # remove the token from histogram
    if histogram is not None:
        histogram.remove_token(iat)
```



결과

Defense	Parameters	Accuracy (%)				Overhead (%)	
		kNN	Pa-SVM [17]	DL-SVM [24]	VNG++ [8]	Latency	Bandwidth
BuFLO [8]	$\tau = 10$s, $\rho = 20$ms, $d = 1500$B	14.9	14.1	18.75	N/A	145	348
CS-BuFLO [2]	$\rho = [20, 200]$ms, $d = 1500$B, CPSP	N/A	30.6	40.5	22.5	173	130
Tamaraw [23]	$\rho_{out} = 0.053$, $\rho_{in} = 0.138, d = 1500$B	13.6	10.59	18.60	12.1	200	38
WTF-PAD	Normal fit, $p = 0.4$, $d = 1500$B	17.25	15.33	23	26	0	54

결론

- 폐쇄 된 환경에서이 방어는 최첨단 공격의 정확도를 91 %에서 20 %로 줄이면서 대기 시간 오버 헤드없이 60 % 미만의 대역폭 오버 헤드를 제공

- 오픈 월드에서는 공격 정확도가 1 %에 불과하며 사이트 수가 증가함에 따라 더 떨어짐

- 추후에는 각 특정 상황에 대한 최적의 히스토그램을 찾기 위해서 유전자 알고리즘으로 최적화

Q & A