[논문리뷰]

IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, VOL. 22, NO. 1, JANUARY/FEBRUARY 2025

MASKCRYPT: Federated Learning With Selective Homomorphic Encryption

Chenghao Hu, *Member, IEEE*, and Baochun Li, *Fellow, IEEE*

PMLC Lab meeting

2025.08.06 (Wed)

MaskCRYPT 논문의 핵심 Research Question

Q. 모델의 **모든 Weight**을 꼭 암호화해야 하는 가?

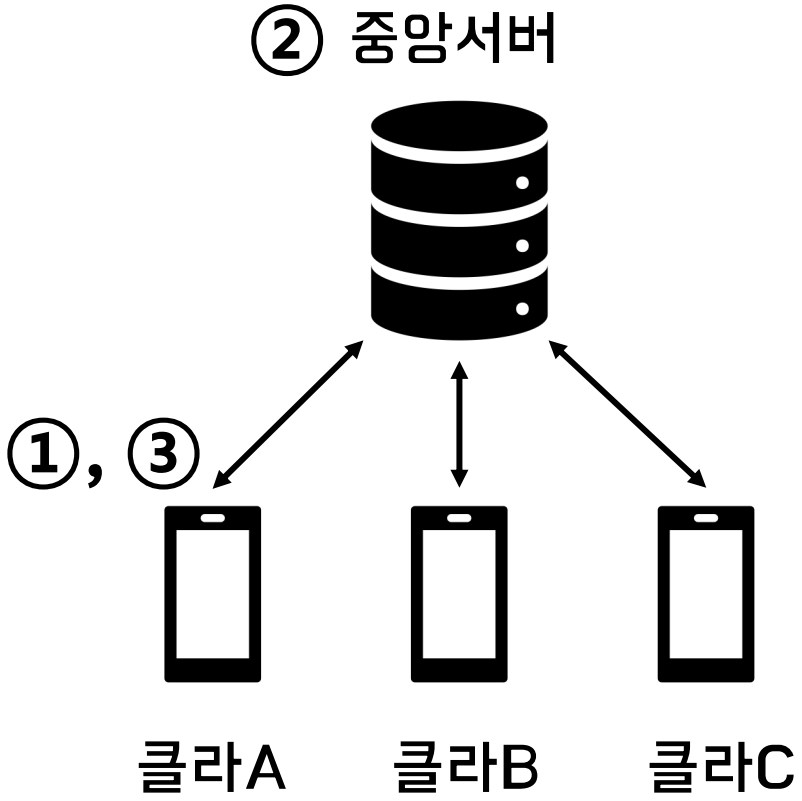
- *원문: Do we have to encrypt all the model weights?*

A. 전체 Weight 중 **일부***만 선택적으로 암호화해도 Robust 방어 가능

* 실험 결과
  : CiFar10, MNIST 데이터셋 기준, 단 1%의 업데이트만 암호화해도
  Membership Inference 및 데이터 복원 공격에 강력한 방어 효과 보임!

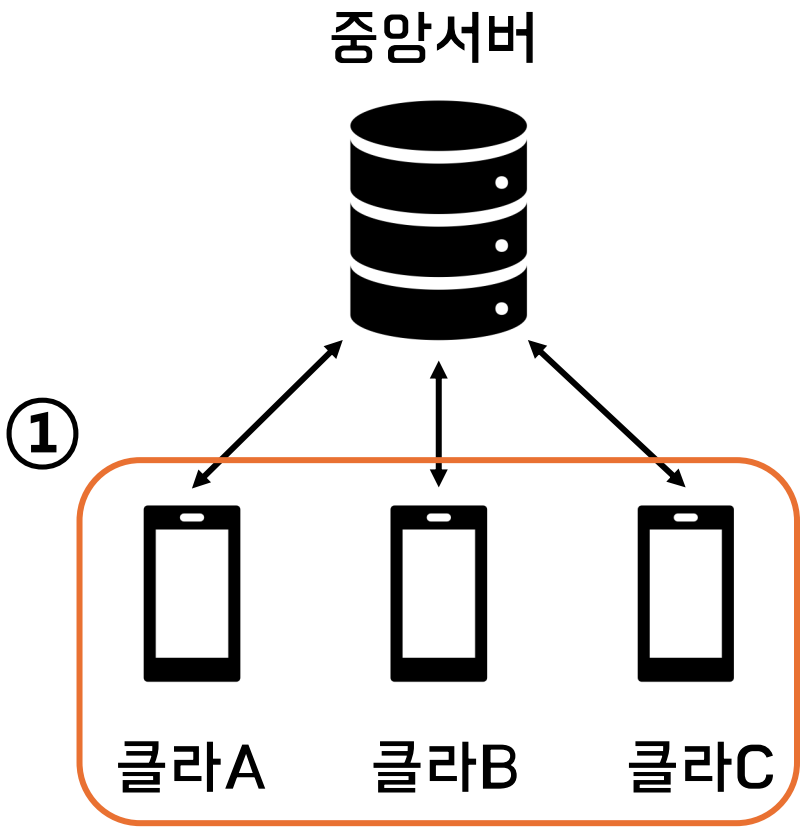
MaskCRYPT의 핵심 Mechanism: *'Selective Homomorphic Encryption'*

1. 각 클라이언트는 **암호화할 Weight Index 우선순위 리스트**를 계산
2. 서버는 클라이언트가 제출한 리스트를 interleave + 중복 제거하여 **공통 마스크 리스트 (Mask Consensus)** 선정
3. 공통 마스크 기반으로 클라이언트는 **동일한 위치의 Weight만 선택적**으로 암호화



MaskCRYPT의 프로세스 (1/3)

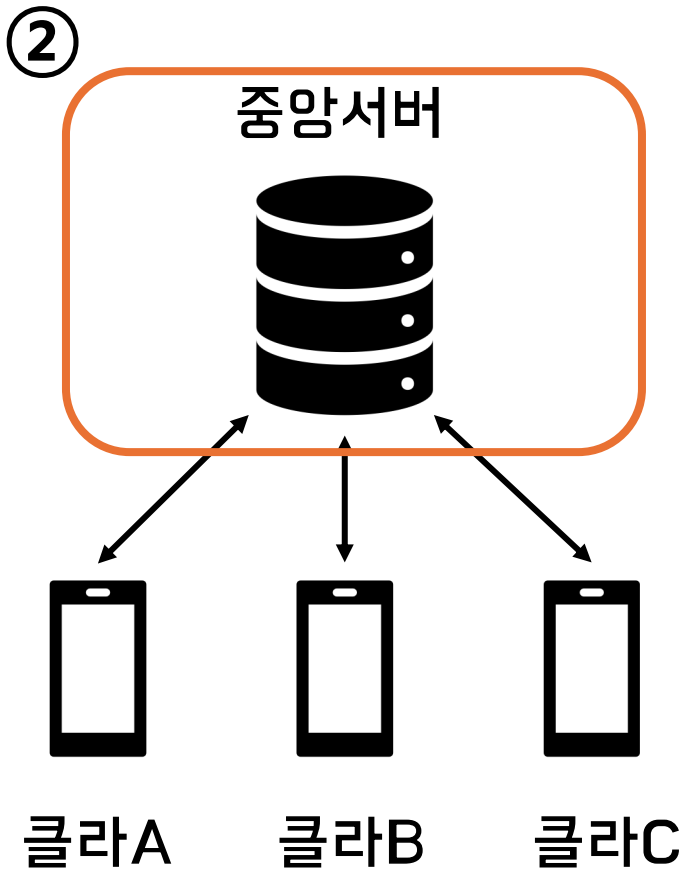
1. 각 클라이언트(클라)는 **자신의 공개키/비밀키 쌍*(p_k, s_k)를 생성**하고, 생성한 공개키를 다른 모든 클라 및 서버와 공유
2. 서버는 초기 글로벌 모델 (w_{t-1})을 모든 클라에게 전달
3. 각 클라는 자신의 로컬 데이터로 글로벌 모델을 학습하여 업데이트된 Weight을 생성



* 공개키/비밀키 쌍(pk, sk)은 동형암호 체계에서 사용하는 키 쌍

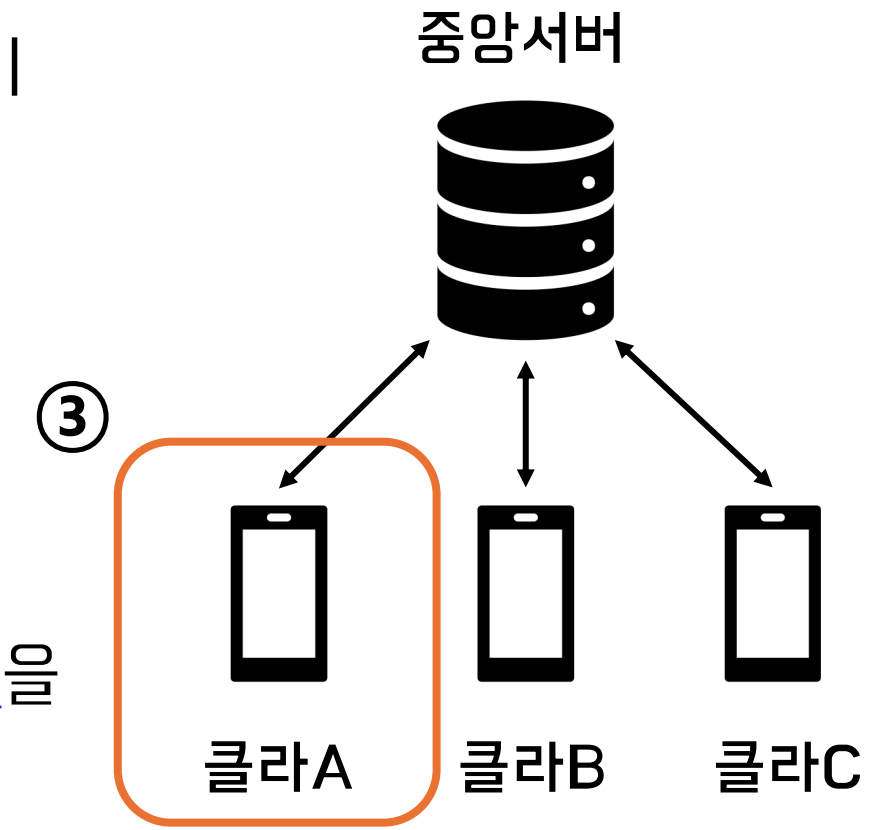
MaskCRYPT의 프로세스 (1/3)

1. 각 클라이언트(클라)는 자신의 공개키/비밀키 쌍(p_k, s_k)를 생성하고, 생성한 공개키를 다른 모든 클라 및 서버와 공유
2. 서버는 **초기 글로벌 모델 (w_{t-1})**을 모든 클라에게 전달
3. 각 클라는 자신의 로컬 데이터로 글로벌 모델을 학습하여 업데이트된 Weight을 생성



MaskCRYPT의 프로세스 (1/3)

1. 각 클라이언트(클라)는 자신의 공개키/비밀키 쌍(p_k, s_k)를 생성하고, 생성한 공개키를 다른 모든 클라 및 서버와 공유
2. 서버는 초기 글로벌 모델 (w_{t-1})을 모든 클라에게 전달
3. 각 클라는 자신의 로컬 데이터로 글로벌 모델을 학습하여 업데이트된 Weight (w_t)을 생성

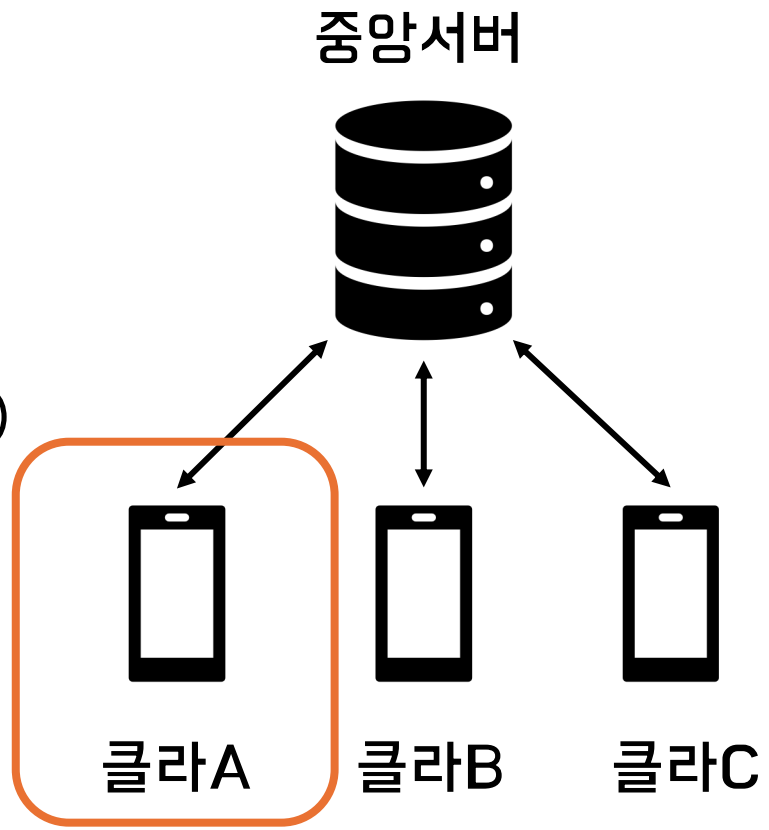


MaskCRYPT의 프로세스 (2/3)

4. 각 클라는 로컬 모델의 학습 결과를 기반으로,
   암호화할 Weight Index 우선순위 리스트를
   계산하고 서버에 전달

5. 서버는 클라로부터 받은 리스트들을 interleave하고
   중복을 제거해 공통 마스크 (Mask Consensus)
   생성한 뒤, 이를 다시 모든 클라에게 전송

6. 각 클라는 공통 마스크에 따라 선택된
   Weight들을 클라 수만큼 균등하게 나누고,
   각 부분을 해당 클라의 공개키로 암호화하여 서버에 전송

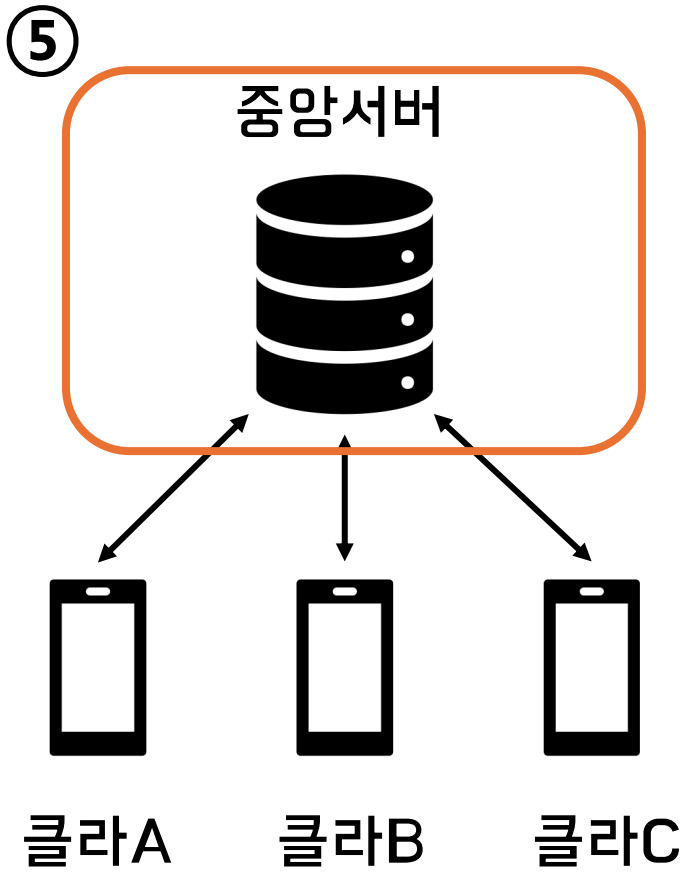


MaskCRYPT의 프로세스 (2/3)

4. 각 클라는 로컬 모델의 학습 결과를 기반으로,
   암호화할 Weight Index 우선순위 리스트를
   계산하고 서버에 전달

5. 서버는 클라로부터 받은 리스트들을 interleave하고
   중복을 제거해 공통 마스크 (Mask Consensus)
   생성한 뒤, 이를 다시 모든 클라에게 전송

6. 각 클라는 공통 마스크에 따라 선택된
   Weight들을 클라 수만큼 균등하게 나누고,
   각 부분을 해당 클라의 공개키로 암호화하여 서버에 전송

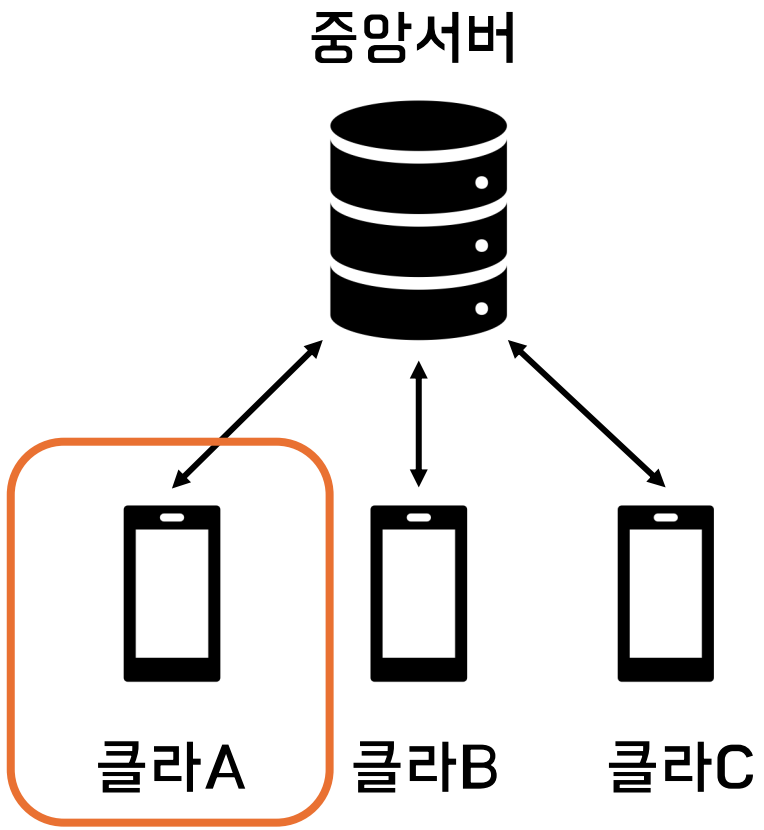


MaskCRYPT의 프로세스 (2/3)

4. 각 클라는 로컬 모델의 학습 결과를 기반으로, 암호화할 Weight Index 우선순위 리스트를 계산하고 서버에 전달

5. 서버는 클라로부터 받은 리스트들을 interleave하고 중복을 제거해 공통 마스크 (Mask Consensus) 생성한 뒤, 이를 다시 모든 클라에게 전송 ⑥

6. 각 클라는 **공통 마스크에 따라** 선택된 Weight들을 클라 수만큼 균등하게 나누고, **각 부분을 해당 클라의 공개키로 암호화***하여 서버에 전송



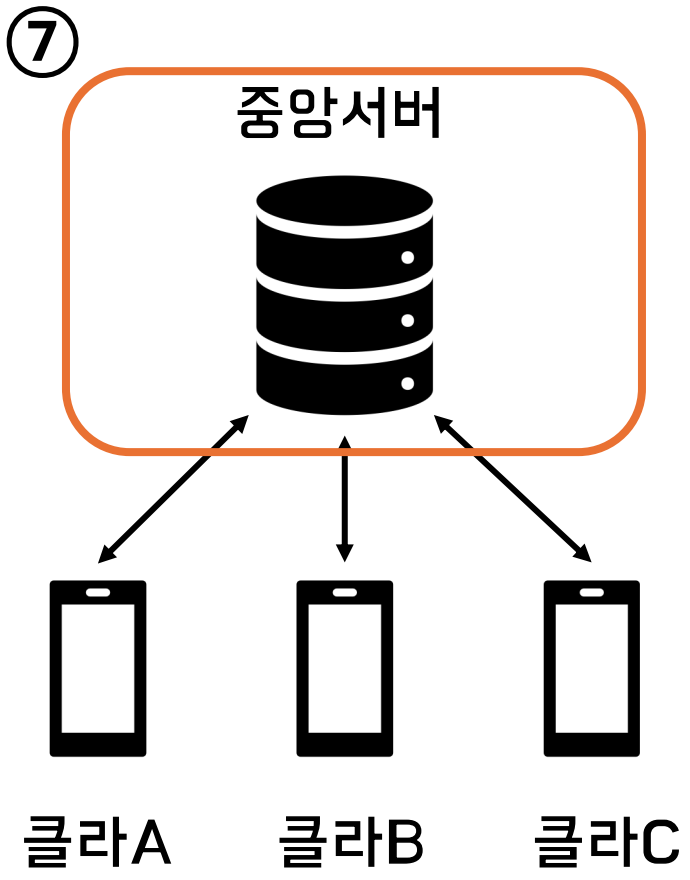
* 공통 마스크에 포함된 Weight 중 일부만 암호화되며, 나머지는 평문으로 서버에 함께 전송됨

MaskCRYPT의 프로세스 (3/3)

7. 서버는 클라로부터 받은 업데이트 중
   ① 평문 Weight은 평문끼리 연산
   ② 암호화된 Weight은 동형암호 연산
   으로 집계한 후 **각 암호문에 해당하는 클라에게 복호화 요청**

8. 각 클라는 서버로부터 받은 암호문을 복호화해 복호화된 값을 서버에 다시 전송

9. 서버는 모든 복호화된 값을 수신한 후 최종 글로벌 모델을 업데이트

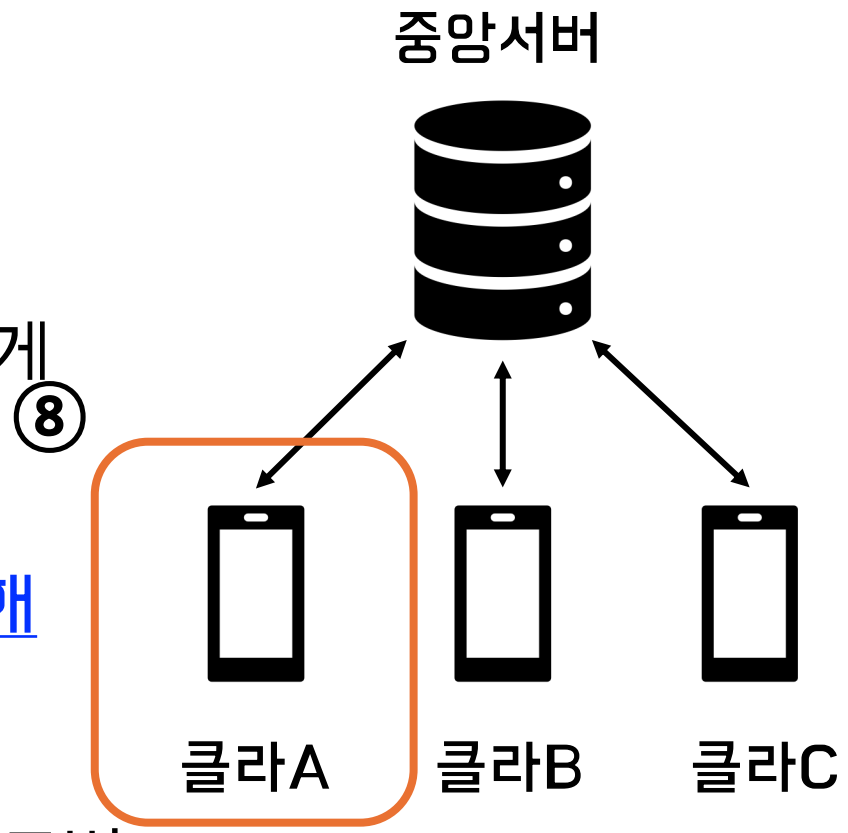


MaskCRYPT의 프로세스 (3/3)

7. 서버는 클라로부터 받은 업데이트 중
   ① 평문 Weight은 평문끼리 연산
   ② 암호화된 Weight은 동형암호 연산
   으로 집계한 후 각 암호문에 해당하는 클라에게 복호화 요청

8. 각 클라는 서버로부터 받은 암호문을 복호화해 복호화된 값을 서버에 다시 전송

9. 서버는 모든 복호화된 값을 수신한 후 최종 글로벌 모델을 업데이트

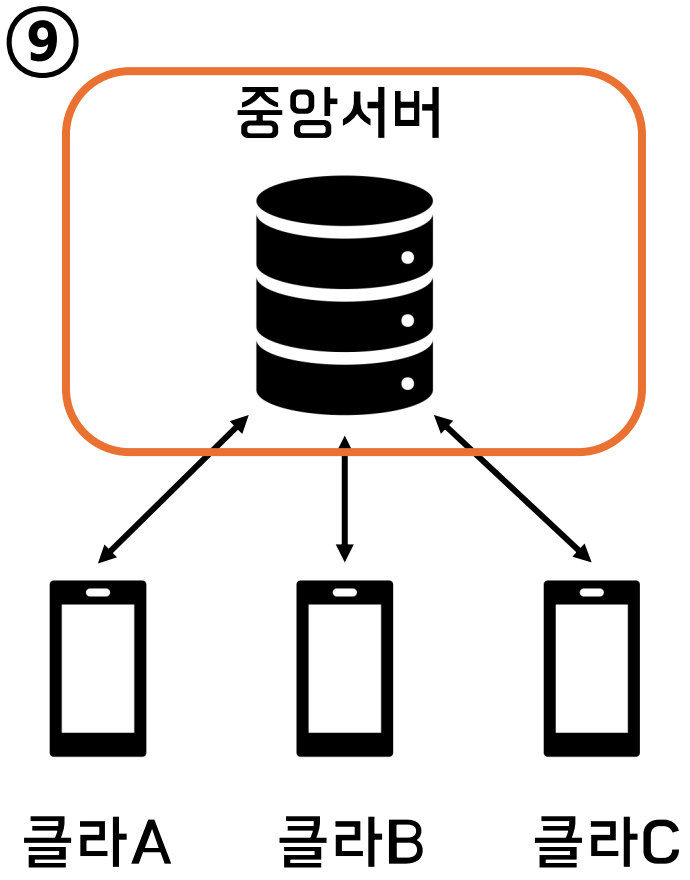


MaskCRYPT의 프로세스 (3/3)

7. 서버는 클라로부터 받은 업데이트 중
   ① 평문 Weight은 평문끼리 연산
   ② 암호화된 Weight은 동형암호 연산
   으로 집계한 후 각 암호문에 해당하는 클라에게 복호화 요청

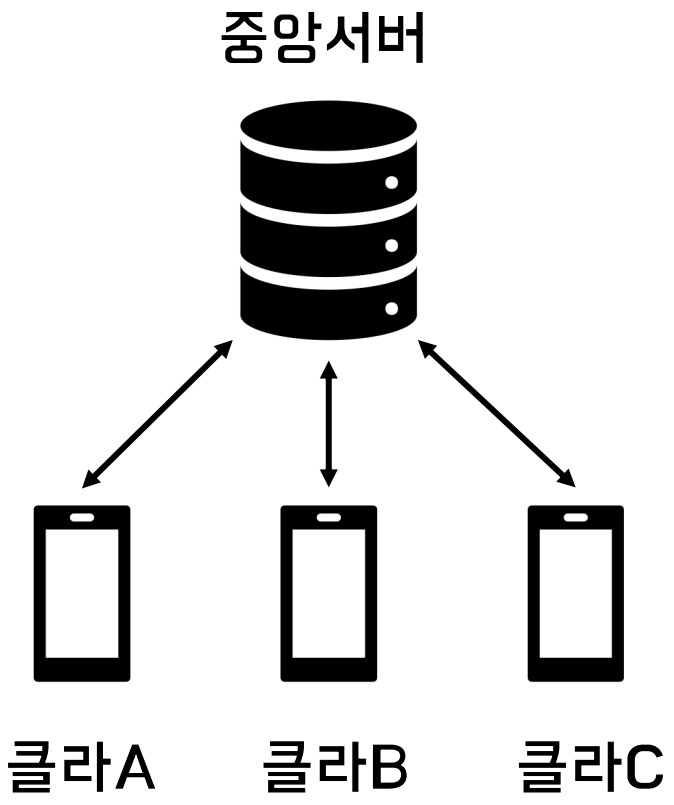
8. 각 클라는 서버로부터 받은 암호문을 복호화해 복호화된 값을 서버에 다시 전송

9. 서버는 모든 복호화된 값을 수신한 후 **최종 글로벌 모델을 업데이트**



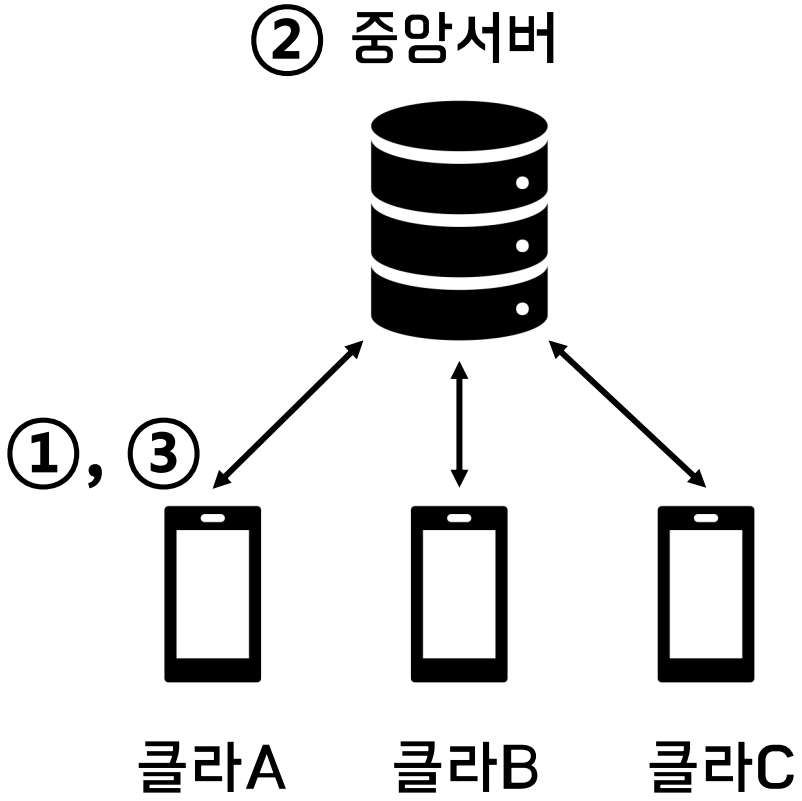
MaskCRYPT의 총 프로세스

1. 각 클라이언트(클라)는 자신의 공개키/비밀키 쌍(pk, sk)를 생성하고, 생성한 공개키를 다른 모든 클라 및 서버와 공유
2. 서버는 초기 글로벌 모델 (w_{t-1})을 모든 클라에게 전달
3. 각 클라는 자신의 로컬 데이터로 글로벌 모델을 학습하여 업데이트된 Weight을 생성
4. 각 클라는 로컬 모델의 학습 결과를 기반으로, 암호화할 Weight Index 우선순위 리스트를 계산하고 서버에 전달
5. 서버는 클라로부터 받은 리스트들을 interleave하고 중복을 제거해 공통 마스크 (Mask Consensus) 생성한 뒤, 이를 다시 모든 클라에게 전송
6. 각 클라는 공통 마스크에 따라 선택된 Weight들을 클라 수만큼 균등하게 나누고, 각 부분을 해당 클라의 공개키로 암호화하여 서버에 전송
7. 서버는 클라로부터 받은 업데이트 중 ① 평문 Weight은 평문끼리 연산 ② 암호화된 Weight은 동형암호 연산으로 집계한 후 각 암호문에 해당하는 클라에게 복호화 요청
8. 각 클라는 서버로부터 받은 암호문을 복호화해 복호화된 값을 서버에 다시 전송
9. 서버는 모든 복호화된 값을 수신한 후 최종 글로벌 모델을 업데이트



(리마인드) MaskCRYPT의 핵심 Mechanism: *'Selective Homomorphic Encryption'*

1. 각 클라이언트는 **암호화할 Weight Index 우선순위 리스트**를 계산
2. 서버는 클라이언트가 제출한 리스트를 interleave + 중복 제거하여 **공통 마스크 리스트 (Mask Consensus)** 선정
3. 공통 마스크 기반으로 클라이언트는 **동일한 위치의 Weight만 선택적**으로 암호화



Selective Homomorphic Encryption (1/3)
: 암호화할 Weight Index 우선순위 리스트 선정 방법

[Gradient-Guided Mask Selection 방법]

- 각 클라는 자신의 데이터로 학습한 후, 어떤 Weight이 가장 민감하게 반응했는지 계산함

- 민감도 = $(w_{t-1}^{*} - w_t) * \text{gradient (e.g., SGD)}$, ($w_{t-1}^{*}$ = 학습 직전의 값)

  → 민감도가 크다는 것은 학습을 통해 많이 바뀌었다 볼 수 있고, 따라서 클라 개별 데이터의 특성을 강하게 반영했을 가능성이 높으므로 암호화 우선 대상 선정!

- 민감도가 큰 상위 ρN개($0 \leq \rho \leq 1, N$은 $Weight$ 개수) 를 암호화 대상 Weight Index로 선정

Selective Homomorphic Encryption (2/3)
: 공통 마스크 리스트 (Mask Consensus) 선정 방법

- 각 클라가 제안한 우선순위 기반 마스크 리스트는 중복된 인덱스를 포함할 수 있음

[Mask Consensus 생성 절차]

① 서버는 각 클라가 보낸 마스크 리스트를 interleave 방식*으로 병합

→ 각 클라의 상위 index들이 고르게 섞이도록 구성

② 병합된 리스트에서 중복을 제거한 뒤, 상위 ρN개($0 \leq \rho \leq 1, N$은 $Weight$ 개수)를 선택하여 최종 Mask Consensus 리스트를 확정

* Interleave 방식은 참고자료에 추가 설명 있음

Selective Homomorphic Encryption (3/3)
: 공통 마스크 (Mask Consensus) 기반 선택적 암호화 방식

- 각 클라는 서버로부터 받은 Mask Consensus 리스트의 인덱스만 선택적으로 암호화 함

[Mask Consensus 기반 선택적 암호화 절차]

① Mask Consensus 리스트(m)를 클라이언트 수 K만큼 균등하게 분할

→ 각 분할된 집단 I_j (j는 클라이언트 구분자) **해당 클라 j의 공개키 p_k 로 암호화**

e.g.) Mask Consensus 리스트 m = {0, 1, 2} → $I_A = \{0\}, I_B = \{1\}, I_C = \{2\}$일 경우,

클라 A의 공개키 p_k 로 0번째 인덱스에 해당하는 데이터 값을 암호화, …

② 암호화가 끝난 후, 각 클라는 평문은 평문끼리, 암호문은 암호문끼리 서버에 전송

MaskCRYPT의 핵심 결과

1. 단 1% weight만 암호화해도,
   ① **Membership Inference Attack 방어 성공**
      - CIFAR10, MNIST 공격 정확도 50%
   ② **데이터 복원 (Reconstruction) 공격 방어 성공**
      - 복원된 이미지 완전히 깨짐

2. 학습 정확도 유지 및 통신 효율성 개선
   ① 모델의 정확도는 기존과 동일하나, 통신량 최대 4.15배 감소
   ② Wall-Clock $Time^*$ 도 감소

* Wall-Clock Time: MaskCrypt에서 진행한 연산시간 + 통신시간 포함한 전체 학습에 걸린 실제 시간

기술적 제약 및 향후 고려사항

1. **클라이언트 공개키 공유 부담**
   - 한계: 모든 클라는 서로의 공개키를 알고 있어야 함
     → 클라 수가 많아질수록 공개키 관리 및 키 유출 위험 증가

2. **복호화 전송 단계의 통신 부담**
   - 한계: 각 클라가 복호화 결과를 서버에 다시 전송해야 함
     → Wall-Clock Time은 줄더라도 여전히 번거로움 존재

3. **공통 마스크 분배 전략의 정당성**
   - 한계: interleave + 중복 제거 방식이 실험적으로 성능이 좋았지만 항상 최적이거나 안전하다는 보장은 없음

Federated Learning 사용시 불편한 점

Federated Learning에서 안전하게

데이터 삭제를 할 수 없을까?

다음 논문 리뷰 대상

Federated Learning의 한계: 데이터 삭제를 안전하게 할 수 없을까?

1. **FedRecovery: Differentially Private Machine Unlearning for Federated Learning Frameworks**
   - 출판: IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, VOL. 18, 2023

2. **Guaranteeing Data Privacy in Federated Unlearning With Dynamic User Participation**
   - 출판: IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, VOL. 22, NO. 3, May/June 2025

3. **Model Recovery in Federated Unlearning With Restricted Server Data Resources**
   - 출판 : IEEE INTERNET OF THINGS JOURNAL, VOL. 12, NO. 11, 1 June 2025

참고자료. Interleave 방식의 3 가지 예시

1. Random 순서 지정
   - 클라 순서를 랜덤하게 설정하기

2. Round-Robin 방식
   - 클라 순서를 순환하며 고르게 분포
     → Round 1: 클라 A → 클라 B → 클라 C
     → Round 1: 클라 B → 클라 C → 클라 A ...

3. Weighted Fairness 방식
   - 클라마다 중요도 또는 가중치를 설정한 뒤, 각자 마스크 리스트에서 추출되는 비율 조정