피앗-샤미르 변환(Theorem 5.1) 증명 분석

이 내용은 앞서 분석한 Theorem 5.1(피앗-샤미르 변환이 안전하다는 정리)의 증명 과정을 설명합니다.

이 증명의 핵심 아이디어는 **귀류법(Reduction)**을 사용하는 것입니다.

> **증명 전략 (귀류법):** "만약 피앗-샤미르로 변환된 **비-상호작용** 프로토콜 $\mathcal{Q}$를 깰 수 있는 효율적인 공격자(P_{FS})가 **존재한다고 가정**하자. 그렇다면, 우리는 이 P_{FS}를 '부품'으로 사용해서, 원본 **상호작용** 프로토콜 $\mathcal{I}$를 깰 수 있는 또 다른 공격자(P^*)를 만들 수 있다. 그런데 원본 $\mathcal{I}$는 안전하다고 전제되었으므로, 이는 모순이다. 따라서 $\mathcal{Q}$를 깰 수 있는 P_{FS}는 존재할 수 없다."

이 증명은 **"또 다른 공격자 P^*를 어떻게 만들 것인가?"**에 대한 구체적인 방법과 성공 확률을 분석합니다.

📄 이미지 내용 상세 설명

1. 공격자 P^*의 설계 (Complete description of P^*)

- **상황:**
  - **P_{FS}**: 우리가 존재한다고 가정한 '나쁜' 공격자입니다. **랜덤 오라클(해시 함수)**을 T번 호출해서 비-상호작용 증명 $\mathcal{Q}$를 깨려고 시도하며, **ϵ**의 확률로 성공합니다.
  - **P^***: 우리가 지금부터 만들 '새로운' 공격자입니다. P^*는 **실제 검증자 V**와 상호작용을 하면서 원본 프로토콜 $\mathcal{I}$를 깨려고 시도합니다. P^*는 P_{FS}를 내부적으로 실행합니다.

- **P^*의 딜레마:** P^*는 P_{FS}에게 가짜 랜덤 오라클(해시 함수) 행세를 해야 합니다. P_{FS}가 오라클에 쿼리((x, α))를 하면, P^*가 응답(β)을 줘야 합니다. P^*는 이 기회를 이용해, P_{FS}의 쿼리 α를 실제 검증자 V에게 보내고, V로부터 받은 실제 챌린지 β를 P_{FS}에게 "이것이 오라클 응답이다"라고 속여서 전달해야 합니다. ***문제는** P_{FS}가 오라클 쿼리를 총 T번 하는데,* P^*는 T번의 쿼리 중 *어느 것이* 진짜 챌린지 β와 연결될 "그 쿼리"인지 모른다는 것입니다.

- **P^*의 전략:**

  1. P^*는 1부터 T까지의 숫자 i를 **무작위로 하나 "추측"(guess)**합니다. (예: "나는 P_{FS}가 날릴 i번째 쿼리가 진짜 챌린지 쿼리일 거라고 확신해!")
  2. P^*는 P_{FS}를 실행시킵니다.
  3. P_{FS}가 1번째부터 $i-1$번째까지 날리는 오라클 쿼리는 P^*가 그냥 임의의 랜덤 값으로 응답해줍니다 (시뮬레이션).
  4. P_{FS}가 **i번째 쿼리** (x, α)를 날리는 순간, P^*는 P_{FS}를 잠시 **일시정지**시킵니다.
  5. P^*는 이 α를 자신이 상대하는 **실제 검증자 V**에게 (원본 프로토콜 $\mathcal{I}$의 첫 번째 메시지로) 전송합니다.
  6. V는 **실제 챌린지 β**를 P^*에게 보냅니다. (이것이 P^*가 이용할 "Public Coin"입니다.)
  7. P^*는 V에게 받은 β를, 일시정지된 P_{FS}에게 "너의 i번째 쿼리에 대한 오라클 응답이다"라며 전달합니다.

8. P_{FS}는 (속았다는 것을 모른 채) $i+1$번째부터 T번째까지 쿼리를 계속합니다 (이것도 P^*가 랜덤 값으로 응답).
9. P_{FS}는 (자신이 ϵ 확률로 성공했다면) V를 속일 수 있는 최종 증명 γ를 출력합니다.
10. P^*는 이 γ를 받아서 V에게 (원본 프로토콜 $\mathcal{I}$의 마지막 메시지로) 전송합니다.

2. P^*의 성공 확률 분석 (Analysis of success probability for P^*)*

- P^*가 V를 속이는 데 성공하려면, **두 가지 조건**이 동시에 충족되어야 합니다.

  1. 애초에 P_{FS}가 성공해야 합니다. (확률: **ϵ**)
  2. P^*가 처음에 "추측"한 i가 P_{FS}가 성공하기 위해 사용한 **바로 그 "올바른 쿼리"**여야 합니다.

- P_{FS}가 성공적인 증명 (α, β, γ)를 만들었다면, T개의 쿼리 중 **정확히 하나**가 $(x, \alpha) \to \beta$에 해당하는 "올바른 쿼리"입니다.

- P^*가 T개 중 그 "올바른 쿼리" 하나를 맞출 확률은 **$1/T$**입니다.

- **결론:** P^*의 최종 성공 확률 = (P_{FS}가 성공할 확률) $\times$ (P^*의 추측이 맞을 확률) $$= \epsilon \times (1/T)$$ $$= \epsilon / T$$

🚀 증명의 의미

- P_{FS}가 비-상호작용 프로토콜 $\mathcal{Q}$를 깰 확률 ϵ이 **무시할 수 없는(non-negligible)** 확률이라고 가정했습니다.
- T는 쿼리 횟수(다항식 시간)이므로 $1/T$도 무시할 수 없는(non-negligible) 값입니다.
- 따라서 P^*의 성공 확률 ϵ / T 역시 **무시할 수 없는(non-negligible) 확률**입니다.
- 이는 P^*가 원본 상호작용 프로토콜 $\mathcal{I}$를 무시할 수 없는 확률로 깼다는 것을 의미합니다.
- 하지만 우리는 $\mathcal{I}$가 "건전성 오류가 무시할 수 있을 만큼(negligible soundness error)" 안전하다고 전제했습니다.
- **모순이 발생했습니다!**

따라서, "$\mathcal{Q}$를 깰 수 있는 P_{FS}가 존재한다"는 최초의 가정이 틀렸습니다. 즉, **$\mathcal{Q}$는 안전합니다.** (정확히는, "계산적으로(Computationally)" 안전합니다.)