

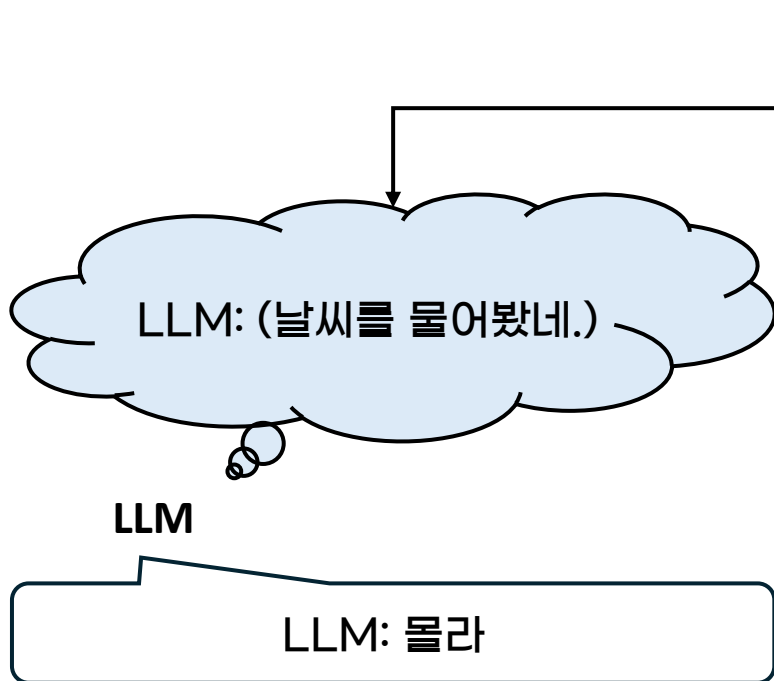
PQC 라이브러리 + AI 에이전트

<https://youtu.be/EXJjh-Bok1Y>

AI 에이전트

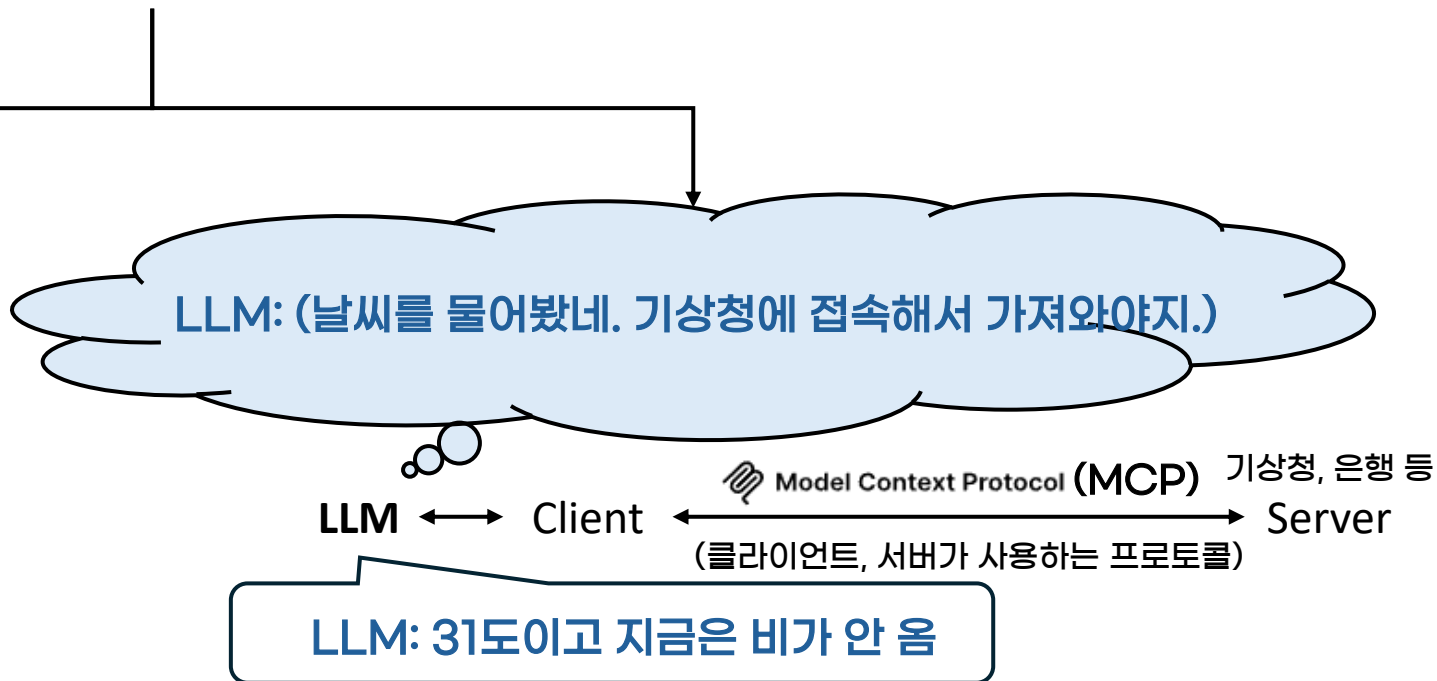
사용자의 목표를 이해하고, 스스로 계획을 세워 다양한 도구(API)를 사용해 실제 작업을 수행하는 지능형 자율 시스템

나: 지금 날씨 어때?



기존 LLM

학습된 데이터 내에서만 답변 가능
→ 외부 정보 필요한 질문 대답 불가



AI 에이전트

질문 파악 후, 외부에서 필요 정보 가져와 답변
미리 설정한 지침에 따라 에이전트를 통해 정보 얻음

AI 에이전트 - 예시

에이전트 X

에이전트 호출하지 않고 지금이 몇 시인지 알려줘

현재 저는 실시간 시각을 확인할 수 없습니다.

대신, 사용 중인 기기나 운영체제의 시계를 확인하시면 정확한 시간을 알 수 있습니다.

에이전트 호출해서 웹 검색하지 말고, 한성대학교 날씨 알려줘



현재 저는 실시간 기상 데이터에 직접 접근할 수 없어서, 웹 검색이나 외부 API 호출 없이 **정확한 현재 날씨**를 알려드릴 수는 없습니다.

다만, 한성대학교는 서울 성북구에 위치해 있으므로, **서울 성북구 평균 기후** 특성을 기반으로 계절별·시간대별 일반적인 날씨 경향은 말씀드릴 수 있습니다.

에이전트 O

지금이 몇 시인지 알려줘

현재 시각은 2025년 8월 10일 13:59 (한국 표준시, KST)입니다.

한성대학교 날씨 알려줘

현재 31° · 화창

서울, 대한민국



오늘

33° 24°

대체로 맑음



월요일

32° 22°

약간 흐림



화요일

34° 25°

대체로 흐리고 더움



수요일

29° 24°

흐림; 오후에 폭우가 내림



목요일

29° 25°

약한 비



금요일

31° 25°

흐리고 습함; 오후에 가벼운 비가 내림



토요일

31° 25°

흐리고 습하며 때때로 소나기가 내림

AI 에이전트와 보안 위협

AI 도메인에서 최근 들어 핫한 주제라고 함

- 튜닝 없이 각자 목적에 맞게 활용할 수 있고 확장성이 굉장히 높음

너무 멋지지만 보안 위협 존재...

- MCP는 JSON/HTTP 기반의 프로토콜을 정의함 → 통신 관련 규약 설정
- 보안 관련 요소도 설정 가능하지만, 여전히 아래와 같은 문제점에 대한 주의 필요
- 프롬프트 인젝션 (악성 명령어 수행), 대리인 문제 (공격자가 권한 남용 가능하도록), 민감 데이터 노출

MCP에서도 보안 위협을 인지하고, 공식 문서에 언급 + 보안 패치 로드맵 제시

2. Attacks and Mitigations

This section gives a detailed description of attacks on MCP implementations, along with potential countermeasures.

Authentication and Security

We're evolving our authorization and security resources to improve user safety and provide a better developer experience:

제안: (인공)지능형보안 인공지능 에이전트

“AI 스스로 맥락에 맞게 상황을 인지하고, 위협을 판단하며, 능동적으로 행동하는 보안 주체가 되도록 함”

- 핵심 기능: AI가 스스로 “지능형 보안 정책 수립”

- AI가 사용자의 프롬프트 문맥을 분석하여 최적의 보안 정책을 수립
 - **요구되는 보안 강도 (민감도) 판단 후**, 그에 맞도록 통신 진행 (**해당 PQC 라이브러리 활용**)

Case 1: 높은 보안 강도 요구

프롬프트: “우리 회사 이번 분기 실적 보고서 분석해줘.”

에이전트가 할 행동: 회사 내부 시스템들에 접근하여 내부 민감 정보 수집

보안 요구 수준 판단: NIST post quantum **level 3 이상 사용**

Case 2: 낮은 보안 강도 요구

프롬프트: “서울 내일 날씨 어때?”

에이전트가 할 행동: 기상청에 접근하여 서울 날씨 수집

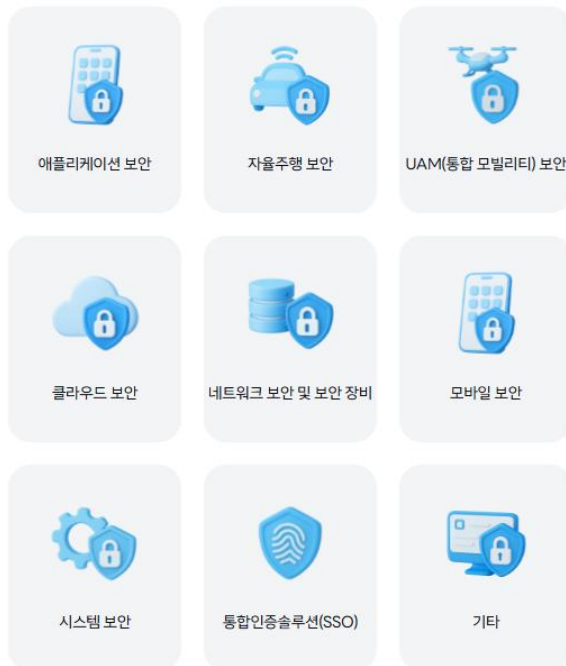
보안 요구 수준 판단: NIST post quantum **level 1 이상 사용**

- AI가 판단한 보안 강도를 만족하는 암호가 서버측에서 지원되지 않는 경우, 해당 추론에 대한 **페널티 부과**
→ **결과에 대한 신뢰성 향상** (위변조 가능성 고려)

PQC 적용 지점

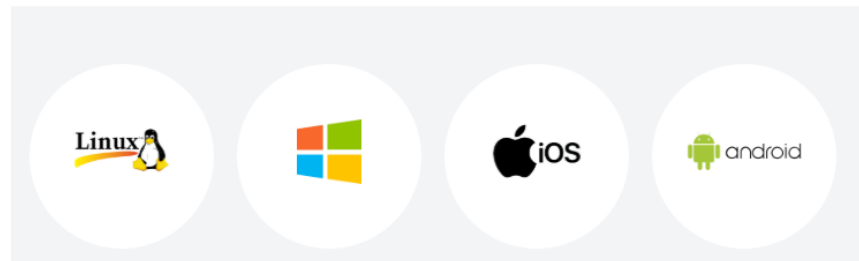
- PQC 적용을 TLS가 아닌 어플리케이션 레벨 보안으로 할 예정
 - 현재 시나리오 (기업 등에서 내부 LLM 사용, 대화형 어플리케이션)에서 TLS는 기본이고 End-to-End 암호화 사용
- 그러나 이 부분의 프로토콜에 대해 자세히 아는 건 없어서 더 살펴볼 예정...

다양한 보안 환경에 PQC 적용



2 다양한 운영 환경 및 확장성

liboqs를 기반으로 다양한 환경에 쉽게 확장 가능한 구조로 개발 (PQC 통합 라이브러리)



출처: 공모전 라이브러리 설명 페이지

3 다양한 보안 프로토콜에 PQC 기술 지원



구현 완료 (case 1)

사용자 프롬프트 질문 (종료=exit): 우리 회사 공지사항 보여줘.

AI (host)의 판단

- 보안 강도 1 써야지

```
[Host] → MCP 호출: finder{'key': '공지사항', 'PQLevel': 'Level_1', 'host_cipherlist': ['Kyber-512', 'Falcon-512']}
```

Server 상황

- 호스트로부터 받은 cipher list
- 서버가 가진 cipher list
- 교집합 (두개 있음)
- Host가 요청하는 보안 강도

```
host_cipherlist: ['Kyber-512', 'Falcon-512']
server_cipherlist: {'Kyber-768', 'Kyber-512', 'Falcon-512'}
intersection(all): ['Falcon-512', 'Kyber-512']
required_strength (from host): 512
```

Server가 반환한 결과

```
[최종 응답]
내일은 쉬는 날 입니다.
[선택된 암호]
Falcon-512
[응답 신뢰도]
1.0
[신뢰도 변경 사유]
satisfy_or_upgrade
```

Host가 요구한 암호를 서버가 가지고 있으므로
해당 암호 사용
→ 신뢰도 1.0 (만점)

구현 완료 (case 2)

사용자 프롬프트

질문 (종료=exit): 미공개재무제표 보여줘.

AI (host)의 판단

- 보안 강도 5 써야지

```
[Host] → MCP 호출: finder{'key': '미공개재무제표', 'PQLevel': 'Level_5', 'host_cipherlist': ['Kyber-1024', 'Falcon-1024', 'Dilithium-5']}
```

Server 상황

- 호스트로부터 받은 cipher list
- 서버가 가진 cipher list
- 교집합 (없음)
- Host가 요청하는 보안 강도

```
host_cipherlist: ['Kyber-1024', 'Falcon-1024', 'Dilithium-5']
server_cipherlist: {'Kyber-768', 'Kyber-512', 'Falcon-512'}
intersection(all): []
required strength (from host): 1024
```

Server가 반환한 결과

```
=====
[최종 응답]
미공개 재무제표는 제공되지 않았습니다.
[선택된 암호]
Kyber-768
[응답 신뢰도]
0.5
[신뢰도 변경 사유]
downgrade_fallback
=====
```

최종 응답은 LLM 성능이 안 좋아서 (파라미터 적은 걸 사용했음)
가져온 결과를 제대로 요약하지 못한 상황일 뿐, 제대로 동작은 함

Host가 요구한 암호를 서버가 가지고 있지 않으므로
같은 암호에서 보안강도 한단계 낮춤
→ 신뢰도 0.5 (사유: 다운그레이드)

사용한 모델

- Google의 Gemma-3, LG의 EXAONE

- LG 공모전이라 EXAONE 사용해봤는데 성능이 괜찮은 것 같음

- Claude와 연동

- 다만, Desktop 앱을 그대로 사용하지 않음 (리눅스 없어서)
- API 사용 (유료)
- 구현했던 MCP 코드 그대로 사용하고 모델만 claude로 변경

항목	Claude Desktop (앱/웹)	Claude API (Anthropic)	
대상	일반 사용자	개발자/엔지니어	
사용 방식	로그인 후 UI에서 대화	API 키 발급 후 코드에서 호출	
비용	무료 플랜 + 월 구독(Pro)	토큰 단위 종량제 과금	
플랫폼	웹, Windows/Mac 앱 (Linux 없음)	어디서든 (Python/Node/REST API)	
MCP 커넥터	지원 (UI에서 설정)	직접 구현해야 함	

- GPT도 시도해봤으나 아직 리눅스를 지원하지 않아서 패스

구현 - 모델 호출

- 모델 사용 및 서빙

- 모델 옵션: 처리할 프롬프트 길이 등에 따라 max_model_len, max_num_seqs 설정 가능

```
tokenizer = AutoTokenizer.from_pretrained(MODEL_DIR, use_fast=True)
llm = VLLM(
    model=str(MODEL_DIR),
    max_new_tokens=1000,
    top_k=10,
    top_p=0.95,
    temperature=0, # 0.5
    tensor_parallel_size=1,
    use_fast=True,
    vllm_kwargs={
        "gpu_memory_utilization": 0.9,
        "max_model_len": 1000,
        "max_num_seqs": 16,
        # "quantization": "bitsandbytes",
        # "load_format": "bitsandbytes",
        "enforce_eager": True,
        "enable_chunked_prefill": True,
    },
)
```

```
def claude_call(prompt: str, max_tokens: int = 800) -> str:
    resp = client.messages.create(
        model=CLAUDE_MODEL,
        max_tokens=max_tokens,
        temperature=0,
        messages=[{"role": "user", "content": prompt}],
    )
```

구현 - 지침 설정

- LLM은 사용자의 프롬프트를 보고, 어떤 데이터를 불러올지 + 어떤 레벨의 PQC를 사용할지를 판단
 - 아래와 같이 LLM이 판단할 때 참고할 룰을 설정함
 - 작업에 맞게 원하대로 변경하면 됨

```
"name": "finder",  
"description": "'key'에 해당하는 우리 회사 문서를 가져오고 적절한 보안 강도를 결정합니다.",
```

```
"key": {{  
  "type": "string",  
  "description": "가져올 문서의 한글 이름 (예: '비상연락망', '비밀파일' 등)"
```

```
"PQLevel": {{  
  "type": "string",  
  "description": "가져올 문서의 한글 이름을 보고 보안 강도를 결정합니다. 선택지는 Level_1, Level_3, Level_5 세 가지입니다.  
  - Level_1: 변조되거나 공개되어도 업무에 심각한 영향을 주지 않는 정보 (예: '사내 행사 일정', '주차장 배정 현황', '공개된 홍보자료')  
  - Level_3: 변조되면 업무에 영향을 주지만, 외부 유출 시 심각한 기밀 손실은 없는 정보 (예: '사원 정보', '내부 매뉴얼', '부서별 실적 요약', '교육자료')  
  - Level_5: 변조되면 안 되고, 외부 유출 시 회사에 심각한 피해를 주는 기밀 정보 (예: '미공개 연구개발 자료', '고객 신상 정보', 'M&A 관련 기밀 문서', '문서명만 보고 위 조건에 따라 정확히 하나의 레벨을 반환하세요."
```

server_dataset

```
≡ Hyunji.txt  
≡ 고객개인정보.txt  
≡ 공지사항.txt  
≡ 미공개재무제표.txt  
≡ 식단표.txt  
≡ 암호구현.txt  
≡ 진행사항보고서.txt
```

호스트가 검색 가능한 데이터들

구현 - host

- Level 1, 3, 5는 LLM이 스스로 판단하여 MCP 호출
- 그 레벨에 따라 사용할 수 있는 암호 알고리즘은 우선 아래와 같이 직접 설정
 - PQC 라이브러리 적용하면서 세부 구현은 조금 달라질 듯

```
if params['PQLevel'] == "Level_1":
    cipherlist = ['Kyber-512', 'Falcon-512']

elif params['PQLevel'] == "Level_3":
    cipherlist = ['Kyber-768']

else:
    cipherlist = ['Kyber-1024', 'Falcon-1024', 'Dilithium-5']
```

```
print(f"[Host] → MCP 호출: {method}{params}")
tool_res_raw = await mcp.call_tool(method, params)
```

```
[Host] → MCP 호출: finder{'key': '공지사항', 'PQLevel': 'Level_1', 'host_cipherlist': ['Kyber-512', 'Falcon-512']}
```

구현 - server

- **호스트는 서버에게 다음 요소들을 보냄**
 - 데이터 요청
 - 요구 보안 레벨
 - 호스트가 사용 가능한 암호 알고리즘
- **서버는 호스트가 사용 가능한 암호 중 선택**
 - 있다면 신뢰도 감소 없이 응답 반환
 - 없다면 호스트가 요구하는 보안 레벨을 기준으로 다음과 같은 옵션 있음
 - 다운그레이드 : 레벨 3을 요구했으나 레벨 1만 사용 가능한 경우, 서버가 가진 레벨 1의 암호 사용 → 응답의 신뢰도 감소
 - 같은 암호 알고리즘에서 레벨만 선택하도록 함
 - 업그레이드 : 응답 신뢰도 감소 없음 (아직 포함 안 함)

```
[선택된 암호]  
Falcon-512  
[응답 신뢰도]  
1.0  
[신뢰도 변경 사유]  
satisfy_or_upgrade
```

```
[선택된 암호]  
Kyber-768  
[응답 신뢰도]  
0.5  
[신뢰도 변경 사유]  
downgrade_fallback
```

계획

- AI 에이전트 적용 옵션은 모두 완료
- PQC 라이브러리 호출은 윤세영이 하고있음
- PQC 라이브러리 활용 관점에서 기능을 추가해야할 것 같음

감사합니다.