제품 제안서 · 2026

설치 없이 동작하는
웹 표준 보안 가상키패드

ActiveX·EXE 설치형 보안 키패드를 완전히 대체하는
웹 표준 기반 SDK입니다. 금융·핀테크·공공기관의
보안 입력 환경을 단 3단계로 전환할 수 있습니다.

AES-256	RSA	3종	Zero
GCM 암호화	OAEP 키 교환	키패드 레이아웃	브라우저 설치

© 2026 coForward · https://vk.coforward.com

PROBLEM & SOLUTION

기존 설치형 키패드의 한계를 넘다

⚠ ActiveX / EXE 의존성

TouchEn nxKey, INISAFE CrossWeb EX 등 기존 솔루션은 Windows IE/Edge 전용 ActiveX 또는 EXE 설치를 강제합니다. Mac·iOS·Android 사용자는 이용 자체가 불가능합니다.

⚠ 모바일·크로스 브라우저 단절

모바일 뱅킹 앱 전환이 필수인 시대에 PC 전용 보안 키패드는 UX 병목이 됩니다. Chrome, Safari, Firefox 완전 지원이 현실적으로 어렵습니다.

⚠ 유지보수 및 배포 비용

OS 업데이트마다 호환성 문제가 반복됩니다. 보안 패치를 전체 클라이언트에 배포하려면 EXE 재설치가 필요하며 운영 비용이 지속 발생합니다.

⚠ 규제 리스크 증가

금융보안원 가이드라인과 OWASP 기준은 비표준 플러그인 사용을 지양하며, 웹 표준 전환을 권고하는 방향으로 지속 강화되고 있습니다.

하이브리드 암호화 — AES-256-GCM + RSA-OAEP

사용자의 입력값은 세션마다 새로 생성된 AES-256-GCM 키로 암호화됩니다. 해당 AES 키는 서버의 RSA-OAEP 공개키로 한번 더 래핑되어 전송되므로, 서버 개인키 없이는 복호화가 불가능합니다.

Canvas + Shadow DOM — DOM에 키값 미노출

모든 키 버튼은 HTML DOM 대신 Canvas API로만 렌더링됩니다. Shadow DOM으로 컴포넌트를 격리하여 외부 스크립트가 `innerHTML` 또는 DOM 탐색으로 키값을 수집할 수 없습니다.

웹 표준 단일 스크립트 — 설치 불필요

외부 의존성 0개. 스크립트 태그 하나만 포함하면 Chrome 80+, Firefox 78+, Safari 14+, Edge 80+ 모든 환경에서 즉시 동작합니다. ActiveX·EXE 설치 안내 페이지가 더 이상 필요 없습니다.

CORE FEATURES

보안과 사용성을 동시에 충족

세션별 무작위 키 배치

Replay & Screenshot Defense

숫자 키패드는 매 세션마다 Fisher-Yates 알고리즘으로 키 배치를 완전 셔플합니다. 화면 캡처나 좌표 기록 기반의 공격을 원천 차단하며, 서버도 배치 순서를 알고 있어 별도 암호화 없이 순서 검증이 가능합니다.

세션 기반 재전송 방지

Anti-Replay Attack

세션 UUID·클라이언트 타임스탬프·nonce를 결합하여 서버에서 검증합니다. 동일한 암호문을 재전송해도 서버가 즉시 거부합니다. 세션 TTL은 기본 300초로 짧게 유지되며 애플리케이션 설정으로 조정 가능합니다.

3종 키패드 레이아웃

Numeric · English · Korean

숫자 키패드(비밀번호), 영문 QWERTY 키패드(ID/패스워드), 한글 두벌식 키패드를 지원합니다. 한글은 자음·모음 조합 로직을 내장해 완성된 문자를 실시간으로 생성합니다.

완전한 CSS 커스터마이징

White-Label Ready

`--coForward-vk-*` CSS 변수로 색상·폰트·크기를 자유롭게 변경합니다. 기관 브랜드 가이드라인에 맞춰 외관을 완전히 커스터마이징할 수 있으며, 7가지 기본 테마(라이트·iOS·Material·다크 계열)가 내장되어 있습니다.

Spring Boot 원클릭 연동

Auto-Configuration

Maven 의존성 추가 후 `@EnableCoForwardVK` 어노테이션 하나로 세션 발급·복호화 API가 자동 활성화됩니다. RSA 키쌍 생성, 세션 저장소, CORS 설정이 모두 자동 구성됩니다.

모든 환경 즉시 지원

Cross-Platform · Zero Install

PC 브라우저(Chrome·Firefox·Safari·Edge), 모바일 브라우저, React·Vue·Svelte 등 모던 프레임워크, 레거시 JSP/HTML 환경 모두 동일한 스크립트 하나로 지원합니다. npm 패키지와 CDN 방식을 모두 제공합니다.

SECURITY ARCHITECTURE

입력부터 서버까지 5단계 보안 레이어



페이지 로드
서버에서 RSA 공개키 & 세션 UUID 수신

사용자 입력
Canvas 전용 렌더링 DOM 미노출 키로거 차단

클라이언트 암호화
AES-256-GCM 입력값 암호화 RSA-OAEP 키 래핑

HTTPS 전송
암호문만 전송 평문 네트워크 노출 없음

서버 복호화
개인키로 AES 키 복원 세션 폐기 처리

🛡 키로거 방지

실제 DOM 키보드 이벤트를 발생시키지 않습니다. Canvas 내부 좌표 클릭만 처리하므로 키로거 소프트웨어가 입력값을 수집할 수 없습니다.

🔒 Shadow DOM 격리

키패드 컴포넌트 전체가 Shadow DOM으로 캡슐화됩니다. 외부 JavaScript가 `document.querySelector`로 내부 상태나 입력값에 접근할 수 없습니다.

♻ 세션 일회성

암호화 세션 UUID는 서버에서 단 한 번만 유효합니다. 소비 후 즉시 폐기되어 동일한 암호문의 재전송 공격을 원천적으로 방지합니다.

📸 스크린샷 방어

숫자 키패드는 세션마다 키 배치가 달라지므로 화면 캡처로 좌표를 기록해도 다음 세션에서는 무효화됩니다.

🔑 HTTPS 필수 강제

RSA 공개키 교환과 암호문 전송은 HTTPS(TLS 1.2+) 환경에서만 동작하도록 설계되었습니다. HTTP 환경에서는 보안이 보장되지 않음을 SDK가 경고합니다.

⏰ 타임스탬프 검증

클라이언트 타임스탬프 허용 오차를 30초로 제한하여 시간 지연 기반 재전송 공격을 차단합니다. 허용 범위는 서버 설정으로 조정 가능합니다.

금융보안원 전자금융감독규정 준수 — 본 SDK는 전자금융거래법 시행령 및 금융보안원의 인터넷뱅킹 보안 가이드라인에서 요구하는 가상 키보드 보안 요건(키값 미노출, 암호화 전송, 무작위 배치)을 모두 충족하도록 설계되었습니다. 실제 도입 전 귀 기관의 내부 보안 담당 부서 검토를 권장합니다.

QUICK START — 3 STEPS

3단계로 즉시 완성되는 보안 키패드

1 프론트엔드 — 스크립트 포함 및 속성 지정

SDK 스크립트를 HTML에 포함하고, 보호할 입력 필드에 data-coForward-vk 속성을 추가합니다. autoInit()이 DOMContentLoaded 이후 자동으로 필드를 감지합니다.

```
<!-- 1. SDK 포함 -->
<script src="dist/coForward-virtual-keypad.js"></script>

<!-- 2. 보호할 필드에 속성 추가 -->
<input type="password" data-coForward-vk="numeric"  placeholder="비밀번호" />
<input type="password" data-coForward-vk="en"  placeholder="영문" />
<input type="password" data-coForward-vk="g4x3"  placeholder="숫자 키보드 4x3 배열" />
<input type="password" data-coForward-vk="g3x4"  placeholder="숫자 키보드 3x4 배열" />
```

2 백엔드 — Maven 의존성 추가

pom.xml에 의존성 내용을 추가합니다.

```
<!-- pom.xml -->
<dependency>
  <groupId>com.coforward.vk</groupId>
  <artifactId>core</artifactId>
  <version>1.0</version>
</dependency>

<dependency>
  <groupId>com.coforward.vk.spring</groupId>
  <artifactId>spring</artifactId>
  <version>1.0</version>
</dependency>
```

3 서버 컨트롤러 — KeypadSessionService로 복호화

KeypadSessionService.submit()을 주입받아 암호화된 요청을 복호화합니다. 세션 ID·암호화 키·IV·데이터를 전달하면 복호화된 평문을 반환합니다.

```
import com.coforward.vk.service.KeypadSessionService;

@Controller
public class SampleController {

    private final KeypadSessionService service;

    public SampleController(KeypadSessionService service) {
        this.service = service;
    }

    @PostMapping("/sample/submit")
    public ResponseEntity<?> submit(@RequestBody Map<String, String> req) {
```

```
        try {
            String plain = service.submit(
                req.get("sessionId"),
                req.get("encrypted_key"),
                req.get("iv"),
                req.get("data")
            );
            return ResponseEntity.ok(Map.of("plain", plain));
        } catch (IllegalArgumentException e) {
            return ResponseEntity.badRequest()
                .body(Map.of("error", e.getMessage()));
        } catch (Exception e) {
            return ResponseEntity.internalServerError()
                .body(Map.of("error", "Internal server error"));
        }
    }
}
```



ActiveX 시대를 끝내고
웹 표준 보안으로 전환하세요

coForward Virtual Keyboard는 금융권 보안 입력의 웹 표준 미래입니다.
개발자 메뉴얼과 라이브 데모를 통해 지금 바로 검토해보세요.

📖 개발자 메뉴얼 | 🌐 라이브 데모

웹사이트
https://vk.coforward.com

라이선스
MIT License v1.0.0

© 2026 coForward Virtual Keyboard

Built with Web Standards — No ActiveX, No Plugin, No Install.