Technical White Paper

임베디드 시스템을 위한 양자 내성 암호 기술의 이해

Walter Schnoor

추상

PQC란 무엇입니까?

PQC(양자 내성 암호)는 기존 컴퓨팅 능력과 암호학적으로 유의미한 양자 컴퓨팅 능력을 모두 보유한 공격자로부터 보호하도록 설계된 암호화 기본 요소의 한 범주입니다.

목차

1 소개 2
2 임베디드 시스템의 보안 목표 2
3 암호화 기본 요소 2
4 CRQC의 잠재적 영향 2
5 미래를 위한 새로운 알고리즘 및 표준 3
6 CNSA 규정 준수 3
7 대칭 암호화 및 해시 함수 3
8 PQC 디지털 서명: ML-DSA 4
9 하드웨어 가속이 필요한가? 4
10 사전 해싱이 필요한가? 6
11 NIST가 선정한 기타 PQC 디지털 서명 체계 6
12 상태 저장형 디지털 서명 체계 7
13 키 합의는 어떻게 되는가? 7
14 미래 대비형 임베디드 시스템 8
15 참고 자료 9

상표

Arm® and Cortex® are registered trademarks of Arm Limited.
모든 상표는 해당 소유권자의 자산입니다.

English Document: STDA060
Copyright © 2026 Texas Instruments Incorporated

1 소개

양자 내성 암호화는 생소하게 느껴질 수 있지만, 미래에 양자 컴퓨팅 기술을 활용하는 공격자에 대비하여 임베디드 시스템을 설계하는 것이 반드시 어렵거나 복잡할 필요는 없습니다. 이는 PQC가 보안을 구현하는 데 사용되는 기반 알고리즘을 변경하지만, 상위 수준의 보안 목표 자체는 변경하지 않기 때문입니다. 즉, '무엇을' 달성할 것인지는 그대로이며, 달라지는 것은 '어떻게' 달성할 것인지입니다. 이 백서에서는 PQC가 임베디드 시스템의 일반적인 보안 목표와 어떻게 연관되는지, PQC 관련 업계 표준은 무엇인지, 그리고 마이크로컨트롤러를 사용하는 리소스 제약이 있는 임베디드 시스템에 PQC가 어떤 영향을 미치는지 설명합니다. 특히 지연, 메모리 및 통신 시간에 미치는 영향을 살펴봅니다.

2 임베디드 시스템의 보안 목표

PQC가 임베디드 시스템에 미치는 영향을 이해하려면 보안 목표 자체부터 살펴보는 것이 가장 좋습니다. 많은 임베디드 시스템은 다음과 같은 공통적인 보안 목표를 가지고 있습니다.

- 보안 부팅은 프로세서에서 실행되는 소프트웨어 또는 펌웨어가 정품인지, 즉 시스템 제조업체가 해당 소프트웨어 또는 펌웨어를 승인했는지 확인합니다.
- 보안 업데이트는 지식 재산권을 보호하기 위해 새로운 소프트웨어 및 펌웨어 업데이트의 기밀성을 유지합니다.
- 증명은 장치 또는 시스템이 비밀 정보를 알고 있음을 입증하여 클라우드 서버나 시스템 내 다른 모듈에 자신이 정품이며 신뢰할 수 있음을 증명할 수 있도록 합니다.
- 보안 통신은 임베디드 시스템 내 장치 간 또는 임베디드 시스템과 원격 서버 간에 주고받는 메시지의 기밀성, 무결성 및 진위성을 유지합니다.

암호화 시스템에 암호화 기본 요소를 적용하면 이러한 보안 목표를 달성할 수 있습니다. 예를 들어 전자 제어 장치의 마이크로컨트롤러는 디지털 서명 알고리즘을 사용하여 애플리케이션 소프트웨어가 CPU에서 실행되도록 허용하기 전에 해당 소프트웨어의 진위성을 검증하는 변경 불가능한 신뢰점 펌웨어를 포함함으로써 보안 부팅을 구현할 수 있습니다. 이를 통해 CPU에서 승인되지 않은 코드가 실행되는 것을 방지할 수 있습니다. 향후 기반이 되는 암호화 기본 요소의 보안이 손상되면 보안 부팅과 같은 보안 목표도 그에 따라 위협받을 수 있습니다.

3 암호화 기본 요소

암호화 기본 요소는 암호화 시스템을 구축하는 데 필요한 암호 기술의 핵심 구성 요소입니다. 암호화 기본 요소의 한 예로 대칭 블록 암호가 있습니다. 대칭 블록 암호는 하나의 비밀 키를 사용하여 평문을 암호문으로 암호화하고, 암호문을 다시 평문으로 복호화합니다. 또 다른 예로 디지털 서명 알고리즘이 있습니다. 디지털 서명 알고리즘은 두 개의 키를 사용하여 메시지의 진위성을 검증합니다. 비밀 키는 메시지에 서명하는 데 사용되며, 공개 키는 해당 메시지가 대응하는 비밀 키로 서명되었는지 검증하는 데 사용됩니다. 이러한 기본 요소의 보안성은 이른바 난해성 가정에 기반합니다.

현재 가장 널리 사용되는 디지털 서명 알고리즘 중 하나는 ECDSA(타원 곡선 디지털 서명 알고리즘)로, ECDLP(타원 곡선 이산 로그 문제)의 난해성을 기반으로 보안성을 확보합니다. 현재의 컴퓨터를 사용하여 ECDLP를 효율적으로 해결할 수 있는 알고리즘은 아직 알려져 있지 않습니다. 따라서 ECDLP는 보안의 기반으로 삼기에 충분히 '난해한' 문제로 간주됩니다. 그러나 양자 컴퓨터의 경우에는 이야기가 다릅니다.

4 CRQC의 잠재적 영향

고전 컴퓨팅은 정의된 연산을 일련의 개별 명령어로 실행하는 오늘날의 컴퓨터를 의미합니다. 경우에 따라 이러한 명령어를 병렬로 실행할 수 있지만, 연산은 순차적 실행 또는 순차적 연산들의 병렬 실행으로 제한됩니다.

반면 양자 컴퓨터는 '0'과 '1'의 상태를 동시에 가질 수 있는 데이터 비트의 특성을 활용하여, 오늘날의 컴퓨터로는 불가능한 방식으로 해를 찾기 위한 탐색 공간을 동시에 탐색할 수 있습니다[1].

CRQC(암호학적으로 유의미한 양자 컴퓨터)는 이론적으로 알려진 알고리즘을 사용하여 ECDLP를 해결할 수 있습니다. 이러한 양자 컴퓨터가 등장한다면 ECDSA와 같이 오늘날 널리 사용되는 암호화 기본 요소의 보안성을 뒷받침하는 난해성에 대한 가정이 무너지게 됩니다. 공격자는 암호화 기본 요소의 보안을 무력화하여 해당 암호화 기본 요소가 뒷받침하는 보안 목표까지 위협할 수 있습니다. ECDSA 공격의 주요 목표 중 하나는 새로운 메시지에 서명하는 데 사용되는 개인 키를 복구하는 것입니다. 공격자는 이러한 개인 키를 사용하여 위조된 메시지에 서명하고 유효한 서명을 생성할 수 있습니다. 이를 위해서는 타원 곡선에서 $P = kG$ 관계가 성립할 때 공개 키 P와 공개 생성점 G를 바탕으로 개인 키 값 k를 찾아야 합니다. 고전 컴퓨터에서는 공개된 값 P와 G를 이용하여 k를 알아내기가 어렵습니다(알려진 방법들은 지수 시간이 소요됨). CRQC를 사용하면 쇼어 알고리즘을 실행할 수 있습니다. 쇼어 알고리즘은 공격에 필요한 시간을 지수 시간에서 다항식 시간으로 단축하여 공격을 실제로 수행할 수 있게 하는 양자 알고리즘입니다. 이 문서 작성 시점에는 NIST(미국 국립표준기술연구소)의 P-256 곡선을 사용하는 ECDSA와 같이 널리 사용되는 ECDSA 구현의 보안을 무력화할 수 있는 쇼어 알고리즘 등의 알고리즘을 실행할 만큼 규모가 크고 안정적인 양자 컴퓨터가 아직 존재하지 않지만, 미래에는 등장할 수 있습니다.

English Document: STDA060
Copyright © 2026 Texas Instruments Incorporated

오늘날 사용되는 기존 암호화 기본 요소를 무력화할 수 있을 만큼 강력한 양자 컴퓨터가 언제 등장할지는 아무도 알 수 없지만 [1], 많은 산업 분야에서 다음과 같은 이유로 미래의 위협에 대비하여 제품의 보안을 강화하기 시작했습니다.

- 신뢰할 수 있는 새로운 암호화 기술을 개발하는 데는 시간이 걸리며, 이를 제품에 통합하는 데도 시간이 필요합니다.
- 많은 임베디드 시스템은 실제 사용 환경에서 10년 이상의 긴 수명을 갖습니다. 지금부터 CRQC의 등장 가능성에 대비하면 향후 보안이 무력화될 위험을 줄일 수 있습니다.
- 특히 엔터프라이즈 및 데이터 센터 시장을 중심으로 다양한 시장에서 규제 도입을 주도하고 있습니다.

그 결과, 보안 목표를 달성하기 위해 PQC 기본 요소로 전환하는 임베디드 시스템 설계자가 늘어나고 있습니다.

5 미래를 위한 새로운 알고리즘 및 표준

대부분의 임베디드 시스템에서 보안 목표는 이를 달성하는 데 사용되는 암호화 기본 요소(기존 암호화 또는 양자 암호화)와 무관합니다. PQC를 도입하는 목적은 임베디드 시스템의 보안 목표를 변경하는 것이 아니라, 해당 보안 목표를 달성하는 데 사용되는 기반 수학적 연산을 변경하여 CRQC를 활용할 수 있는 공격자로부터 난해한 문제를 보호하는 것입니다.

6 CNSA 규정 준수

미국에서는 국가 안보 시스템에 사용되는 제품이 NSA(미국 국가안보국)의 CNSA(상용 국가안보 알고리즘) 제품군을 준수해야 합니다. 2022년 NSA는 CNSA 제품군을 버전 1.0에서 버전 2.0으로 업데이트했으며, CNSA 2.0의 주요 도입 배경 중 하나는 PQC를 통합하는 것이었습니다. CNSA 제품군을 새로 도입하는 경우 2027년 1월 1일까지 버전 2.0을 준수해야 하며, 이미 구축된 시스템은 2030년 12월 31일까지 버전 2.0을 준수하거나 CNSA 2.0을 준수하는 시스템으로 교체해야 합니다.

표 6-1에서는 CNSA 1.0에서 승인된 암호화 기본 요소와 CNSA 2.0에서 승인 및 요구되는 암호화 기본 요소를 비교하여 나열합니다.

표 6-1. CNSA 1.0과 CNSA 2.0에서 승인된 암호화 기본 요소 비교

애플리케이션 사용	암호화 기본 요소	CNSA 1.0 승인됨	CNSA 2.0 승인됨[2]
범용 암호화	대칭 암호화	FIPS(연방 정보 처리 표준) PUB(간행물) 197 (AES[고급 암호화 표준]-256만 해당)	
	보안 해시	FIPS PUB 180-4 (보안 해시 알고리즘[SHA]-384만 해당)	FIPS PUB 180-4 (SHA-384 또는 SHA-512)
	디지털 서명	FIPS PUB 186-4 (ECDSA P-384 또는 리베스트, 샤미르, 애들먼 3,072비트 모듈러스)	FIPS PUB 204 (모듈 격자 기반 디지털 서명 표준[ML-DSA]-87만 해당)
	키 협상	타원 곡선 디피-헬만 P-384 또는 유한 필드 디피-헬만(3,072비트 모듈러스)	FIPS PUB 203 (ML-KEM[모듈 격자 기반 키 캡슐화 메커니즘]-1,024만 해당)
펌웨어 또는 소프트웨어 서명에만 사용	디지털 서명	-	NIST SP(특별 간행물) 800-208 (LMS[레이턴-미칼리 서명] 또는 XMSS[확장 머클 서명 체계][1])
하드웨어 무결성 검사에만 사용	보안 해시	-	FIPS PUB 202 (SHA-3-384 또는 SHA-3-512)

7 대칭 암호화 및 해시 함수

CNSA 2.0에서는 대칭 암호화 기본 요소와 보안 해시 함수에 대한 최소 요구 사항이 크게 변경되지 않았습니다. 예를 들어 CNSA 2.0에서는 대칭 암호화에 대해 CNSA 1.0에서 사용하던 것과 동일한 알고리즘 및 보안 수준(AES-256)을 유지합니다. 이는 CRQC가 현재 AES-256에 심각한 위협이 되지 않기 때문입니다. AES의 보안을 낮출 수 있는 양자 알고리즘이 존재하지만, CNSA 1.0과 2.0에서 채택한 256비트 키 크기는 충분히 크기 때문에 AES는 여전히 안전합니다. 그로버 알고리즘을 예로 들어 보겠습니다. 그로버 알고리즘은 AES에 대한 무차별 대입 공격에 필요한 시간을 2k에서 2k/2로 단축할 수 있는 양자 알고리즘으로, 여기서 k는 비트 단위의 키 크기를 나타냅니다. 그로버 알고리즘을 사용하는 최상의 조건에서도 AES-256의 보안 강도는 2^{256}에서 2^{128}으로 감소하는 수준이며, 이는 여전히 안전한 것으로 간주됩니다. 반면 AES-128을 사용하는 설계에서는 보안 강도가 2^{128}에서 2^{64}으로 감소하여 취약해질 수 있습니다. 따라서 현재 AES-128을 사용하는 애플리케이션에서는 AES-256으로 전환하여 양자 컴퓨터의 이점을 무력화하는 것이 권장됩니다.

CNSA 2.0에서 가장 중요한 변경 사항 중 일부는 디지털 서명 및 키 합의와 관련이 있습니다.

Copyright © 2026 Texas Instruments Incorporated

8 PQC 디지털 서명: ML-DSA

펌웨어 및 소프트웨어 업데이트 서명을 포함한 범용 디지털 서명에 대해 CNSA 2.0에서는 ML-DSA를 사용하도록 요구합니다. ML-DSA는 모듈 오류 학습 문제의 계산적 난해성을 기반으로 보안성을 확보하며, 고전 컴퓨팅 및 양자 컴퓨팅 능력을 모두 갖춘 공격자에 대해서도 안전한 것으로 여겨집니다. ML-DSA는 NIST에 제출된 Crystals-Dilithium 서명 체계를 기반으로 설계되었으며, NIST는 세 차례의 분석 및 평가를 거쳐 이를 FIPS 204 표준화 대상으로 선정했습니다[4].

보안 부팅, 증명 및 기타 서명 사용 사례를 포함하여 디지털 서명에 ECDSA를 사용하는 임베디드 애플리케이션에서 ML-DSA는 유사한 프로그래밍 인터페이스를 제공하는 양자 내성 보안 대안입니다. ML-DSA는 확장성도 갖추고 있으며, 키 및 서명 크기, 연산 지연, 연산에 필요한 SRAM(정적 임의 액세스 메모리) 공간, 보안 간의 절충 관계에 따라 세 가지 보안 수준(ML-DSA-44, ML-DSA-65 및 ML-DSA-87)을 지원합니다. CNSA 2.0에서는 가장 높은 보안 수준인 ML-DSA-87을 의무화하고 있습니다.

임베디드 시스템 개발자 관점에서 ML-DSA는 다음과 같은 측면에서 ECDSA와 상당한 차이가 있습니다.

- ML-DSA는 ECDSA보다 키와 서명의 크기가 훨씬 큽니다. ML-DSA-87의 서명 크기(4,627바이트)는 ECDSA P-384(96바이트)보다 48배 이상 크기 때문에, 증명 기능을 수행하거나 새로운 소프트웨어 및 펌웨어 업데이트를 설치할 때 더 많은 플래시 및 SRAM을 확보해야 하며, 유선 전송 시간도 더 길어집니다.
- CPU 기반 구현을 비교할 때 ML-DSA 검증은 ECDSA 검증보다 빠른 경향이 있어 인증 시간을 단축할 수 있습니다. 예를 들어 ML-DSA-87의 서명 검증 연산은 NIST P-384 곡선을 사용하는 ECDSA에 비해 10분의 1의 시간으로 수행할 수 있어 마이크로컨트롤러의 부팅 시간을 크게 단축할 수 있습니다.
- ML-DSA 구현에서는 일반적으로 필요한 연산을 수행하는 데 ECDSA보다 더 많은 SRAM이 필요하며, 작업 메모리와 실행 지연 사이에 다양한 절충 관계가 존재합니다.

표 8-1에서는 ML-DSA의 각 보안 수준별로 기존 컴퓨팅 서명(ECDSA P-384)과 양자 내성 컴퓨팅 서명(ML-DSA)을 비교합니다.

표 8-1. ECDSA와 ML-DSA의 매개 변수 크기 및 성능 비교

매개 변수	ECDSA	ML-DSA-44	ML-DSA-65	ML-DSA-87
보안 수준	P-384	카테고리 2	카테고리 3	카테고리 5
비밀 키(바이트)	48	2,570	4,032	4,896
공개 키(바이트)	96	1,312	1,952	2,592
서명(바이트)	96	2,420	3,309	4,627
서명 전송 시간(초)(1)	0.009	0.22	0.30	0.42
키 생성 시간(초)(2)	1.12	0.084	0.14	0.23
서명 시간(초)(2)	1.23	≅0.4	≅0.8	≅1.6
검증 시간(초)(2)	**2.26**	**0.10**	**0.16**	**0.26**
검증에 필요한 SRAM(킬로바이트)(2)	2.2	21.9	29.1	37.6

(1) 100kHz 속도에서 바이트당 9비트를 기준으로 한 서명 전송 시간
(2) 테스트 환경은 TI MSP 암호화 라이브러리를 실행하는 TI(텍사스 인스트루먼트)의 80MHz Arm® Cortex®-M0+ 마이크로컨트롤러입니다

9 하드웨어 가속이 필요한가?

타원 곡선 암호화를 사용하는 임베디드 시스템을 개발하는 개발자는 애플리케이션이 지연에 민감한 경우 타원 곡선 연산을 가속하는 기능이 탑재된 마이크로컨트롤러를 사용할 수 있습니다. 그러나 이로 인해 애플리케이션에 실제로 필요한 것보다 더 비싼 마이크로컨트롤러를 선택하게 될 수 있습니다.

예를 들어 챌린지-응답 애플리케이션에서 80MHz Arm Cortex-M0+ 프로세서를 사용하여 챌린지-응답 체계의 ECDSA P-384 서명을 계산하는 데 2초 이상이 걸릴 수 있습니다. 이는 100Kbps(표준 I^2C 또는 115200 직렬 범용 비동기 리시버 트랜스미터와 같은 통신 방식을 가정한 일반적인 속도)로 생성된 서명을 전송하는 데 걸리는 6ms보다 훨씬 긴 시간입니다.

그러나 PQC로 전환하면 디지털 서명의 크기가 커지고 연산 시간이 짧아지면서 이러한 상황이 크게 달라집니다. 예를 들어 서명을 전송하는 데 필요한 회선 시간은 ECDSA P-384의 약 9ms에서 ML-DSA-87의 약 420ms로 약 48배 증가합니다. 서명 검증에 필요한 연산 시간은 ECDSA P-384의 약 2,200ms에서 약 260ms로 단축되며, 이는 디지털 서명을 전송하는 데 걸리는 시간의 거의 절반에 해당합니다. 따라서 증명과 같은 애플리케이션은 연산 시간보다 통신 시간에 더 큰 제약을 받는 경향이 있으므로 CPU 기반 구현이 경제적입니다. 그림 9-1에서는 Arm Cortex-M0+ 및 Arm Cortex-M33 대상 MCU에서 각 보안 수준별 검증 성능을 비교합니다.

Copyright © 2026 Texas Instruments Incorporated

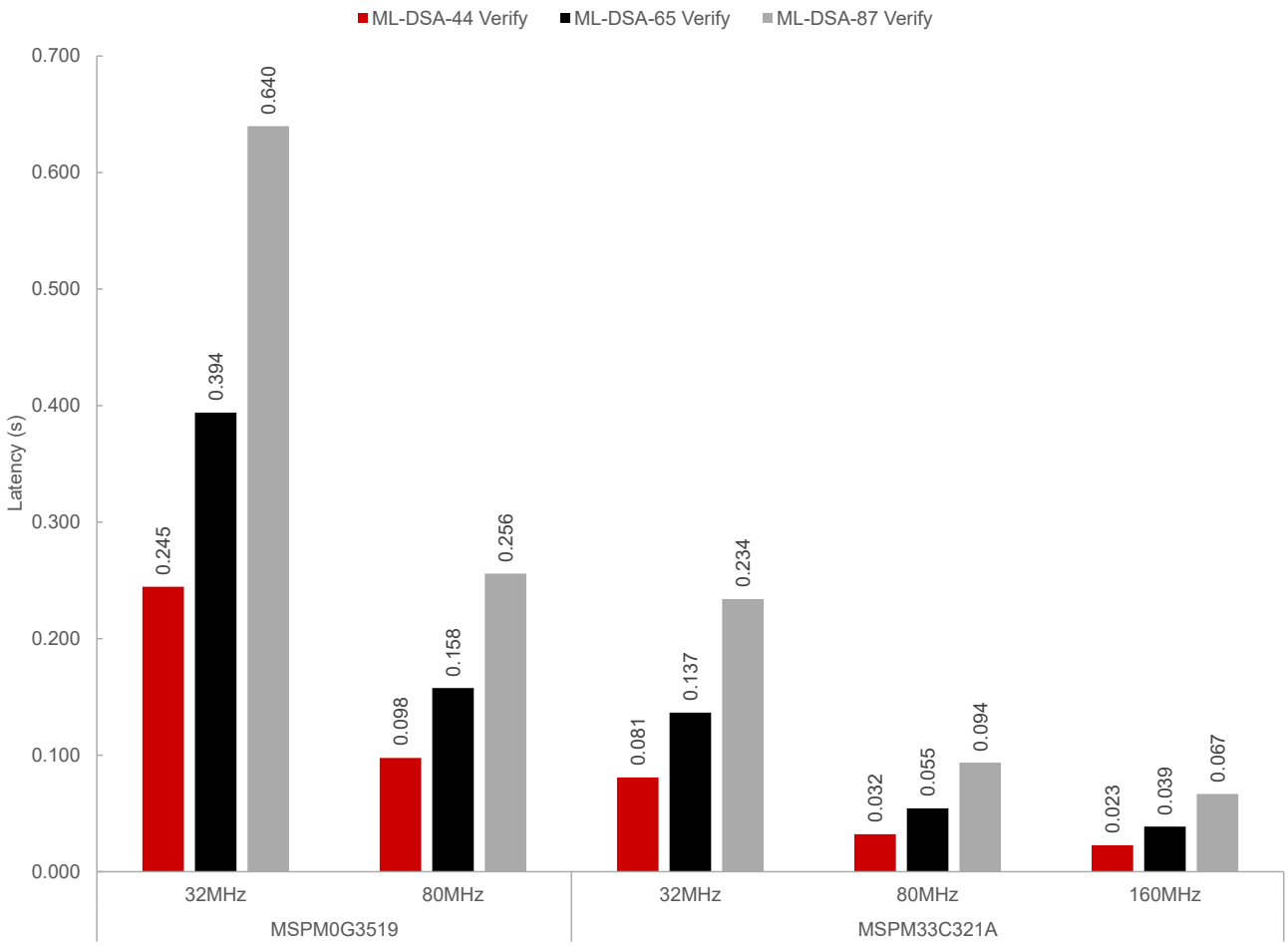


그림 9-1. TI MSPM0 및 MSPM33 마이크로컨트롤러의 ML-DSA 검증 성능

Copyright © 2026 Texas Instruments Incorporated

10 사전 해싱이 필요한가?

보안 부팅 구현에서는 일반적으로 먼저 SHA-256 또는 SHA-384와 같은 보안 해시 함수를 사용하여 애플리케이션 소프트웨어 및 펌웨어 이미지의 해시값을 계산하고, 이를 통해 고정 크기의 메시지 대푯값을 생성한 다음 ECDSA를 사용하여 서명하고 검증합니다. ML-DSA는 이러한 '사전 해싱' 단계를 없앱니다. 실제로 ML-DSA 설계자들은 이 체계를 사용하는 사용자가 디지털 서명을 생성할 때 메시지의 전체 내용을 서명 알고리즘에 직접 입력하는 것을 선호합니다.

FIPS PUB 204에서는 이 프로세스를 "순수" 모드라고 합니다. FIPS PUB 204에서는 HashML-DSA라는 사전 해싱 모드도 정의하고 있습니다. 이 모드에서는 서명 생성 연산에 사용된 해시 함수에 관한 정보를 포함하여 해당 해시 함수를 서명에 바인딩합니다.

FIPS 204 외에도 세 번째 옵션을 사용할 수 있습니다. 이는 FIPS 204의 사전 해싱 모드인 HashML-DSA를 사용하는 대신, 메시지를 사전에 해싱한 다음 해당 해시값을 ML-DSA에 직접 전달한 메시지처럼 취급하여 순수 모드의 ML-DSA로 서명하는 것입니다. 이 방식에서는 해시 함수가 서명에 바인딩되지 않지만, 현재 ECDSA를 SHA와 함께 사용하는 방식과 더 유사합니다.

ML-DSA는 디지털 서명을 생성하고 검증할 때 내부적으로 Keccak 함수를 사용합니다. Keccak은 SHA-3 함수와 SHA 및 Keccak(Shake) 확장 가능 출력 함수 내부에서 사용되는 핵심 함수입니다. ML-DSA는 메시지의 보안 해시 값 생성과 검증 과정에서의 공개 키 결정적 확장 등 다양한 용도로 Keccak을 사용합니다.

이는 리소스가 제한된 시스템에서 중요한 의미를 갖습니다. ML-DSA를 포함하면 Keccak도 함께 포함되므로 동일한 코드 공간 내에서 보안 해시 함수를 제공합니다. 따라서 ML-DSA의 순수 모드를 사용하면 SHA-384 또는 SHA-512와 같은 추가 해시 함수를 저장하기 위해 비휘발성의 코드 메모리 공간을 추가로 할당할 필요가 없습니다. 또한 Keccak은 SHA-3를 애플리케이션의 다른 용도로 사용할 수 있는 범용 해시 함수로 제공할 수 있습니다.

현재 CNSA 2.0을 준수하는 시스템에서는 승인된 보안 해시 함수(SHA-384 또는 SHA-512)를 사용하여 메시지를 사전에 해싱할 수 있지만, FIPS PUB 204에 명시된 ML-DSA의 사전 해싱 모드(HashML-DSA)는 사용할 수 없습니다. 대신 FIPS PUB 204의 순수 모드가 기본으로 사용됩니다.

11 NIST가 선정한 기타 PQC 디지털 서명 체계

NIST는 ML-DSA 외에도 두 가지 양자 내성 디지털 서명 체계를 표준화 대상으로 선정했습니다.

- Sphincs+ 서명 방식을 기반으로 하는 SLH-DSA(무상태 해시 기반 디지털 서명 표준)는 FIPS PUB 205로 표준화되었습니다.
- FN-DSA(N차 절단 다항식 환에서의 고속 푸리에 변환을 이용한 단위 격자 기반 디지털 서명 알고리즘), 즉 Falcon은 현재 FIPS PUB 206으로 표준화가 진행 중입니다.

SLH-DSA는 해시 기반 서명 체계입니다. ML-DSA와 마찬가지로 무상태 방식입니다. ML-DSA와 달리 SLH-DSA는 보안 수준이 높을수록 서명 크기가 매우 커져 수십 킬로바이트에 이르므로, 임베디드 애플리케이션, 특히 마이크로컨트롤러 기반 애플리케이션에서 활용하는 데 제약이 있습니다. NIST가 SLH-DSA를 표준화했음에도 불구하고, 현재 CNSA 2.0에서는 SLH-DSA 기반 서명을 허용하지 않습니다.

FN-DSA는 복잡성이 증가하는 대신 ML-DSA 및 SLH-DSA보다 더 작은 서명을 제공할 것으로 기대됩니다. FN-DSA는 부동 소수점 연산에 의존하는데, 이러한 연산은 프로세서 아키텍처 전반에서 일정한 시간으로 구현하기가 매우 어려운 것으로 알려져 있습니다. 또한 모든 임베디드 아키텍처에 부동 소수점 유닛이 탑재되어 있는 것은 아니므로, 성능상의 이점이 상쇄될 수 있습니다. 현재 CNSA 2.0에서는 FN-DSA를 허용하지 않으며[3], NIST도 아직 FN-DSA를 표준화하지 않았습니다.

이러한 이유로 업계에서는 성능, 크기 및 보안성 간의 균형이 우수한 ML-DSA를 사실상 디지털 서명 체계로 채택하는 추세입니다.

Copyright © 2026 Texas Instruments Incorporated

12 상태 저장형 디지털 서명 체계

ML-DSA와 앞서 설명한 서명 체계는 모두 무상태 체계입니다. 특정 개인 키를 사용하여 서명할 때 상태를 유지할 필요가 없다는 점은 임베디드 애플리케이션에서 선호되는 특성입니다. 특히 리소스가 제한된 환경에서 정상 작동 중 챌린지-응답 서명이 빈번하게 수행되는 증명 사용 사례에서 유용합니다.

무상태 서명 체계가 표준화되기 전에는 NIST SP 800-208에 표준화된 LMS 및 XMSS를 비롯한 여러 양자 내성 상태 저장형 서명 체계가 소프트웨어 및 펌웨어 이미지 서명(보안 부팅)과 같은 특정 사용 사례에 대한 옵션으로 제공되었습니다.

이러한 서명 체계는 ML-DSA보다 작은 공개 키를 제공하므로, 새로운 펌웨어의 진위성을 검증할 때 필요한 임베디드 비휘발성 메모리(일반적으로 플래시 메모리)의 용량을 줄일 수 있어 보안 부팅을 구현하는 임베디드 시스템에 유리합니다. 다만 서명 작업에서는 개인 키의 기밀성을 유지하기 위해 해당 개인 서명 키의 상태를 추적해야 합니다. 펌웨어 및 소프트웨어 업데이트는 빈번하게 이루어지지 않으며, 통제된 IT 환경에서는 서명용 하드웨어 보안 모듈의 상태를 유지할 수 있으므로 이러한 방식이 허용될 수 있습니다. 그러나 임베디드 환경에서는 이러한 방식이 허용되지 않는 경우가 많습니다.

따라서 CNSA 2.0을 준수하는 시스템에서는 LMS와 XMSS를 사용할 수 있지만, 펌웨어 및 소프트웨어의 디지털 서명 용도로만 허용되며 증명과 같은 다른 사용 사례로는 사용할 수 없습니다. 반면 ML-DSA는 보안 부팅과 증명을 비롯한 모든 디지털 서명 사용 사례로 사용할 수 있습니다.

13 키 합의는 어떻게 되는가?

키 합의는 안전하지 않은 통신 채널을 통해 두 개체 간에 공유 비밀을 설정하는 데 필요한 단계입니다. 디지털 서명과 마찬가지로 FFDH(유한체 디피-헬만) 또는 ECDH(타원 곡선 디피-헬만)와 같이 오늘날 널리 사용되는 암호화 기본 요소에 의존하는 키 합의 알고리즘도 쇼어 알고리즘을 이용한 양자 기반 공격에 취약합니다. 이러한 문제를 해결하기 위해 NIST는 FIPS 203에서 ML-KEM(모듈 격자 기반 키 캡슐화 메커니즘)을 표준화했습니다. ML-KEM은 현재 FFDH 또는 ECDH가 사용되는 환경에서 활용하도록 설계된 양자 내성 KEM(키 캡슐화 메커니즘)입니다.

핸드셰이크 과정에서 양측이 생성한 난수로부터 공유 비밀을 도출하는 디피-헬만 방식의 설계와 달리, KEM은 키 생성, 캡슐화 및 역캡슐화 단계로 구성됩니다. 클라이언트-서버 모델에서 ML-KEM을 사용한 키 합의 프로세스는 다음과 같습니다.

1. 먼저 서버에서 키 쌍을 생성합니다. 캡슐화 (또는 '공개') 키를 클라이언트에 배포합니다. 역캡슐화(또는 '개인') 카는 서버에서 비공개로 유지합니다.
2. 클라이언트가 서버와 보안 통신 채널을 설정하고 대칭 키에 합의하려는 경우, 난수 시드를 기반으로 ML-KEM 캡슐화 연산을 수행하여 암호문과 공유 비밀을 얻습니다. 그런 다음 암호문을 서버로 전송합니다.
3. 서버는 암호문을 수신하고 역캡슐화 연산을 수행하여 클라이언트와 동일한 공유 비밀을 얻습니다.
4. 이제 서버와 클라이언트는 공유 비밀 키를 사용하여 통신할 수 있습니다.

ML-KEM은 소프트웨어로 구현해도 높은 성능을 발휘할 수 있으며, 160MHz Cortex-M33 MCU에서 캡슐화 시간이 최단 7ms에 불과합니다. 그림 13-1에서는 Arm Cortex-M0+ 및 Arm Cortex-M33 대상 MCU에서 각 보안 수준별 캡슐화 성능을 비교합니다.

Copyright © 2026 Texas Instruments Incorporated

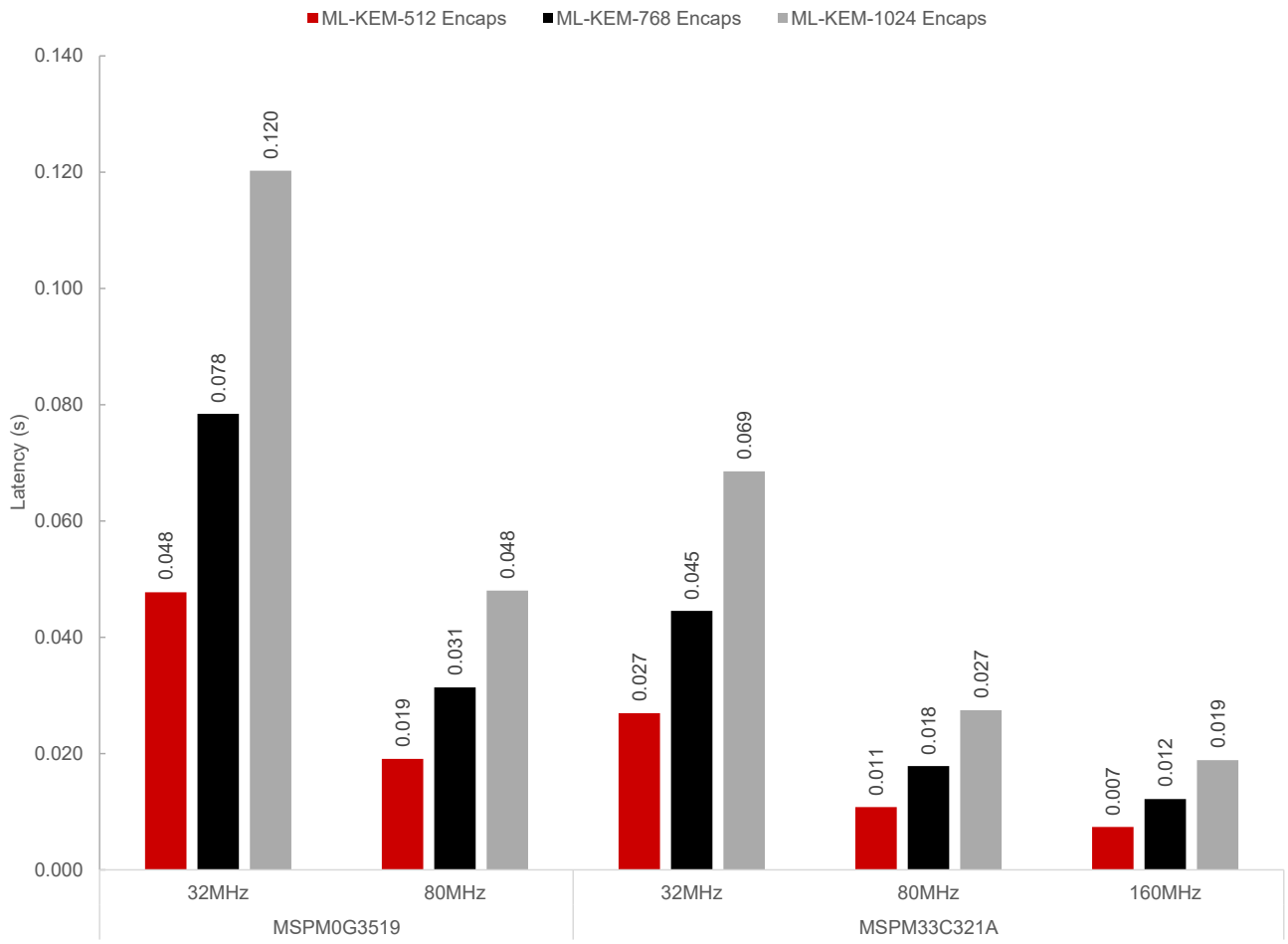


그림 13-1. TI MSPM0, MSPM33 마이크로컨트롤러의 ML-KEM 캡슐화 성능

ML-KEM은 양자 내성을 갖도록 설계되었다는 점 외에도 기타 유용한 보안 특성을 제공합니다. 특히 IND-CCA2(선택 암호문 공격에 대한 구별 불가능성)를 지원합니다. 이는 공격자가 클라이언트에서 서버로 전송되는 암호문을 캡슐화와 역캡슐화 사이에 조작할 수 있더라도, 역캡슐화 결과를 무작위 값과 구별할 수 없으므로 공격에 유리한 정보나 추가적인 지식을 얻을 수 없다는 의미입니다. 또한 ML-KEM의 출력은 대칭 암호화 키로 직접 사용할 수 있으므로 임베디드 애플리케이션의 구현을 간소화하고 추가적인 키 도출 단계가 필요하지 않을 수도 있습니다.

14 미래 대비형 임베디드 시스템

설계자는 향후 공격자가 보안 부팅, 증명 및 보안 통신과 같은 기본적인 보안 목표를 달성하는 데 사용되는 기존 암호화 기본 요소를 공격할 수 있을 만큼 강력한 양자 컴퓨터를 보유하게 될 가능성을 고려해야 합니다. 미국은 NIST와 NSA를 통해 이러한 능력을 갖춘 공격자에 대해서도 안전할 것으로 여겨지는 암호화 기본 요소를 선정하기 위한 프레임워크를 마련했습니다.

임베디드 시스템의 과제로는 작은 비휘발성 메모리 및 작업 메모리 용량, 느린 통신 인터페이스, 제한적인 상태 정보 저장 능력 등이 있습니다. TI는 초저전력 및 저비용 마이크로컨트롤러에서 양자 내성 암호화 기본 요소를 지원하여 CNSA 2.0 규정을 준수하는 동시에 성능과 메모리 간의 균형을 유지할 수 있도록 합니다. 여기에는 다음이 포함됩니다.

- 디지털 서명: 사전 해싱 및 순수 모드에서 ML-DSA-44, ML-DSA-65 및 ML-DSA-87을 완벽하게 지원합니다.
- 키 캡슐화: ML-KEM-512, ML-KEM-768 및 ML-KEM-1024를 완벽하게 지원합니다.
- 대칭 암호화: 암호 블록 체인 메시지 인증 코드가 결합된 카운터 모드 및 갈루아 카운터 모드를 비롯한 AES-256과 관련 작동 모드를 지원합니다.
- 보안 해시 함수: SHA-2, SHA-3 및 SHAKE 제품군을 완벽하게 지원합니다.

Copyright © 2026 Texas Instruments Incorporated

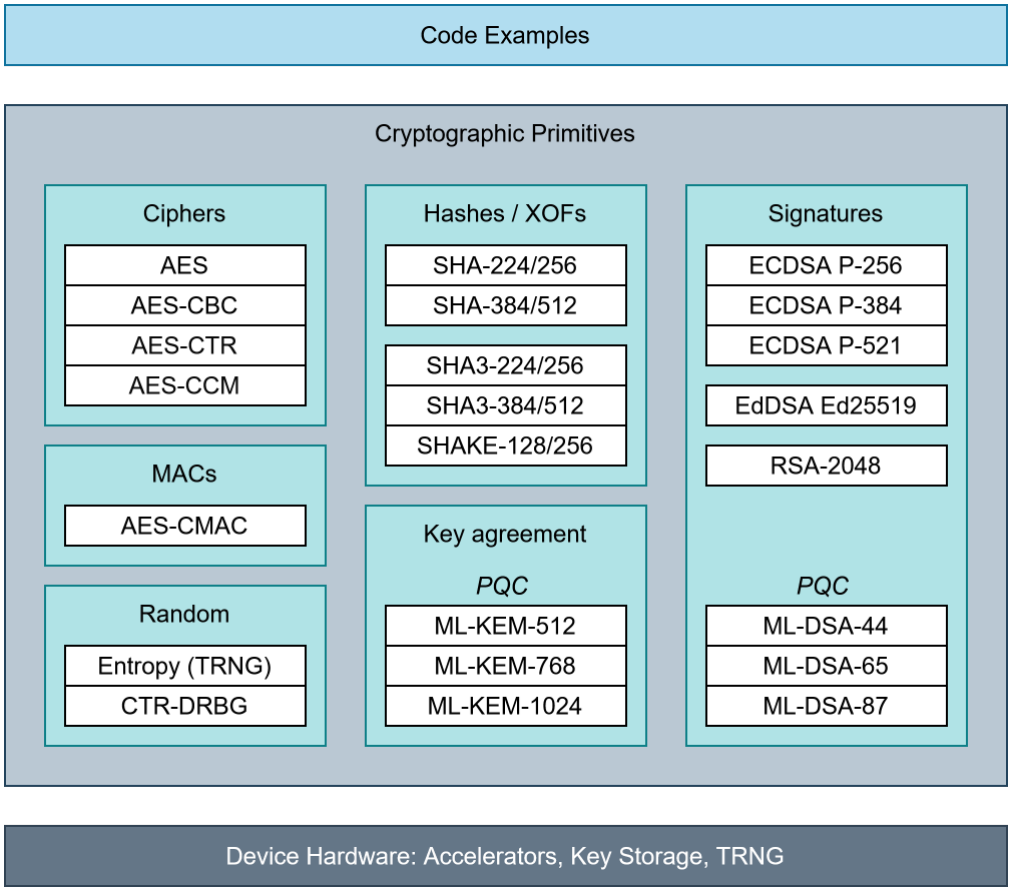


그림 14-1. MSP-CRYPTO-LIB 기능

Arm Cortex-M0+ 기반 MSPM0 및 Arm Cortex-M33 기반 MSPM33 마이크로컨트롤러에서 TI MSP 암호화 라이브러리를 사용하면 보안 부팅 사용 사례를 위한 CNSA 2.0 준수 ML-DSA-87 디지털 서명 검증을 260ms(Cortex-M0+) 또는 70ms(Cortex-M33) 이내에 수행할 수 있습니다. 이를 통해 리소스와 비용이 제한된 환경에서도 빠른 부팅 시간을 제공하는 동시에 양자 컴퓨팅이 현실화될 미래에 대비할 수 있습니다.

15 참고 자료

1. NIST. 2024년 8월 13일. "양자 내성 암호화란 무엇인가?" 2026년 2월 3일에 액세스.
2. NSA. 2022년 9월. "상용 국가안보 알고리즘 제품군 2.0 발표" 2026년 2월 3일에 액세스.
3. NSA. 2024년 12월. "상용 국가안보 알고리즘 제품군 2.0 및 양자 컴퓨팅 관련 FAQ" 2026년 2월 3일에 액세스.
4. NIST FIPS 204. 2024년 8월 13일. "모듈 격자 기반 디지털 서명 표준." 2026년 6월 6일 액세스함.

Copyright © 2026 Texas Instruments Incorporated

중요 알림 및 고지 사항

TI는 기술 및 신뢰성 데이터(데이터시트 포함), 디자인 리소스(레퍼런스 디자인 포함), 애플리케이션 또는 기타 디자인 조언, 웹 도구, 안전 정보 및 기타 리소스를 "있는 그대로" 제공하며 상업성, 특정 목적 적합성 또는 제3자 지적 재산권 비침해에 대한 묵시적 보증을 포함하여(그러나 이에 국한되지 않음) 모든 명시적 또는 묵시적으로 모든 보증을 부인합니다.

이러한 리소스는 TI 제품을 사용하는 숙련된 개발자에게 적합합니다. (1) 애플리케이션에 대해 적절한 TI 제품을 선택하고, (2) 애플리케이션을 설계, 검증, 테스트하고, (3) 애플리케이션이 해당 표준 및 기타 안전, 보안 또는 기타 요구 사항을 충족하도록 보장하는 것은 전적으로 귀하의 책임입니다.

이러한 리소스는 예고 없이 변경될 수 있습니다. TI는 리소스에 설명된 TI 제품을 사용하는 애플리케이션의 개발에만 이러한 리소스를 사용할 수 있는 권한을 부여합니다. 이러한 리소스의 기타 복제 및 표시는 금지됩니다. 다른 모든 TI 지적 재산권 또는 타사 지적 재산권에 대한 라이선스가 부여되지 않습니다. TI는 이러한 리소스의 사용으로 인해 발생하는 모든 청구, 손해, 비용, 손실 및 책임에 대해 책임을 지지 않으며 귀하는 TI와 그 대리인을 완전히 면책해야 합니다.

TI의 제품은 TI의 판매 약관 , TI의 일반 품질 지침또는 ti.com 이나 해당 TI 제품과 함께 제공되는 기타 조건의 적용을 받습니다. TI가 이러한 리소스를 제공한다고 해서 TI 제품에 대한 TI의 해당 보증 또는 보증 부인 정보가 확장 또는 기타의 방법으로 변경되지 않습니다. TI가 명시적으로 제품을 사용자 정의 또는 고객 정의용으로 지정하지 않는 한, TI 제품은 범용의 표준 카탈로그 장치입니다.

TI는 사용자가 제안할 수 있는 어떠한 추가적이거나 상이한 조건도 반대하며 이를 거부합니다.

Copyright © 2026, Texas Instruments Incorporated

최종 업데이트: 2025/10/25

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, regulatory or other requirements.

These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you fully indemnify TI and its representatives against any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI’s products are provided subject to TI’s Terms of Sale, TI’s General Quality Guidelines, or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI’s provision of these resources does not expand or otherwise alter TI’s applicable warranties or warranty disclaimers for TI products. Unless TI explicitly designates a product as custom or customer-specified, TI products are standard, catalog, general purpose devices.

TI objects to and rejects any additional or different terms you may propose.

Copyright © 2026, Texas Instruments Incorporated

Last updated 10/2025