参考資料

Technical White Paper

組込みシステムにおける耐量子暗号の基礎

Walter Schnoor

概要

PQC とは？

耐量子暗号 (PQC) は、古典コンピューティングの能力と、暗号解読能力を持つ量子コンピューティングの能力の両方を備えた攻撃者から保護するために設計された、暗号プリミティブの一種です。

目次

1 概要....2
2 組込みシステムのセキュリティ目標....2
3 暗号プリミティブ....2
4 CRQC の潜在的な影響....2
5 新しいアルゴリズムと将来の標準....3
6 CNSA への準拠....3
7 対称暗号化とハッシュ関数....3
8 PQC デジタル署名：ML-DSA....4
9 ハードウェア アクセラレーションは必要ですか？....5
10 プリハッシュは必要ですか？....6
11 NIST が選定したその他の PQC デジタル署名方式....6
12 ステートフル デジタル署名方式....7
13 鍵合意はどうでしょうか？....7
14 組込みシステムの将来への備え....8
15 参考資料....9

商標

Arm® and Cortex® are registered trademarks of Arm Limited.
すべての商標は、それぞれの所有者に帰属します。

Copyright © 2026 Texas Instruments Incorporated

1 概要

耐量子暗号は難解に聞こえるかもしれませんが、量子コンピュータを使う将来の攻撃者に備えて組込みシステムを対策することは、決して難しいことではありません。PQC では、セキュリティの実現に使う基盤のアルゴリズムは変わりますが、上位レベルのセキュリティ目標は変わらないからです。つまり、「何を」実現するかは同じで、変わるのは「どのように」実現するかです。このホワイト ペーパーでは、次の内容について説明します。PQC が組込みシステムの一般的なセキュリティ目標とどう関わるか、PQC に関連する業界標準、そしてマイコンを使うリソースに制約のある組み込みシステムへの影響です。この影響とは、レイテンシ、メモリ、通信時間への影響のことです。

2 組込みシステムのセキュリティ目標

PQC が組込みシステムに与える影響を理解するには、セキュリティ目標そのものから考えるのが最善です。多くの組込みシステムには、次のような共通のセキュリティ目標があります:

- プロセッサで実行されるソフトウェアやファームウェアが正規のもの、つまりシステムの製造元が承認したものであることを保証するセキュア ブート。
- 知的財産を保護するために、新しいソフトウェアやファームウェアの更新の機密性を保つセキュア アップデート。
- デバイスやシステムが秘密情報を知っていることを示し、クラウド サーバーやシステム内のほかのモジュールに対して、自身が正規のものであり信頼できることを証明するアテステーション。
- 組込みシステム内のデバイス間、または組込みシステムとリモート サーバー間でやり取りされるメッセージの機密性、完全性、真正性を保つセキュアな通信。

これらのセキュリティ目標は、暗号システムに暗号プリミティブを導入することで実現します。たとえば、電子制御ユニット内のマイコンでは、変更不可能なルート オブ トラスト ファームウェアを組み込むことでセキュア ブートを実装できます。このファームウェアは、アプリケーション ソフトウェアを CPU で実行する前に、デジタル署名アルゴリズムを使ってソフトウェアを認証します。これにより、承認されていないコードが CPU で実行されるのを防止します。基盤となるプリミティブが将来破られると、セキュア ブートなどのセキュリティ目標も、それに伴って損なわれる可能性があります。

3 暗号プリミティブ

暗号プリミティブとは、暗号システムを構築するのに必要な、暗号の中核となる構成要素です。暗号プリミティブの一例が対称ブロック暗号です。対称ブロック暗号は、一つの秘密鍵を使って平文を暗号文に暗号化し、暗号文を平文に復号します。もう一つの例がデジタル署名アルゴリズムです。デジタル署名アルゴリズムでは、二つの鍵を使ってメッセージの真正性を検証できます。メッセージに正規のものとして署名するための秘密鍵と、対応する秘密鍵でメッセージが署名されたことを検証するための公開鍵です。これらのプリミティブの安全性は、困難性の仮定と呼ばれるものに基づいています。

現在最も広く使われているデジタル署名アルゴリズムの一つが、楕円曲線デジタル署名アルゴリズム (ECDSA) であり、この安全性は、楕円曲線離散対数問題 (ECDLP) の困難性に基づいています。現時点では、現在のコンピュータで ECDLP を効率的に解くアルゴリズムは知られていません。そのため、ECDLP は安全性の根拠とするのに十分「困難」であると考えられています。しかし、量子コンピュータは別の話です。

4 CRQC の潜在的な影響

古典コンピューティングとは現在のコンピュータのことで、定義された演算を一連の離散的な命令として実行します。状況によってはこれらの命令を並列化できますが、演算は逐次処理か、逐次処理を並列に組み合わせたものに限られます。

一方、量子コンピュータは「0」と「1」の両方の状態を同時にとれるデータ ビットを利用できます。そのため、現在のコンピュータでは不可能な方法で解空間を同時に探索できます [1]。

暗号解読能力を持つ量子コンピュータ (CRQC) は、理論上、既知のアルゴリズムを使って ECDLP を解くことができます。このような量子コンピュータが実現すれば、ECDSA など現在広く使われている暗号プリミティブの安全性を支える困難性の仮定が崩れます。攻撃者は暗号プリミティブを破ることができ、その結果、その暗号プリミティブで実現しているセキュリティ目標も損なわれます。ECDSA に対する攻撃の主な目的は、新しいメッセージを正当なものとして署名するのに使う秘密鍵を復元することです。秘密鍵を手に入れた攻撃者は、不正なメッセージに署名して有効な署名を作成できます。そのためには、公開鍵 P と公開生成点 G から、秘密鍵の値 k を求める必要があります。ここで、楕円曲線上で $P = kG$ です。古典コンピュータでは、公開値 P と G から k を復元するのは困難です (既知の手法では指数時間がかかります)。CRQC を使えば、Shor のアルゴリズムを実行できる可能性があります。これは、攻撃にかかる時間を指数時間から多項式時間に短縮し、攻撃を現実的なものにする量子アルゴリズムです。本稿の執筆時点では、Shor のアルゴリズムなど、

Copyright © 2026 Texas Instruments Incorporated

ECDSA の一般的な実装 (米国国立標準技術研究所 [NIST] の P-256 曲線を使う ECDSA など) を破るアルゴリズムを実行できるほど大規模で安定した量子コンピュータはまだ存在しません。ただし、将来は実現する可能性があります。

現在の古典暗号プリミティブを破れるほど強力な量子コンピュータがいつ実現するかは誰にもわかりません [1]。それでも多くの業界が、次の理由から、将来の脅威に備えて製品のセキュリティ対策を始めています:

- 信頼できる新しい暗号の開発には時間がかかり、それを製品に組み込むのにも時間がかかります。
- 多くの組込みシステムは、現場で 10 年以上寿命が延長されています。CRQC が実現する可能性に今から対処しておけば、将来セキュリティが失われるリスクを低減できます。
- 多くの市場、特にエンタープライズ市場とデータセンター市場が規制を推進しています。

その結果、セキュリティ目標を達成するために PQC 暗号プリミティブへ移行する組み込みシステム設計者が増えています。

5 新しいアルゴリズムと将来の標準

ほとんどの組み込みシステムのセキュリティ目標は、その実現に使う暗号プリミティブ (古典暗号か耐量子暗号か) には左右されません。PQC を導入する目的は、組込みシステムのセキュリティ目標を変えることではありません。セキュリティ目標の実現に使う基盤の演算を変えて、CRQC の能力を持つ攻撃者に対しても困難性問題が守られるようにすることです。

6 CNSA への準拠

米国では、国家安全保障システムで使われる製品は、国家安全保障局 (NSA) の商用国家安全保障アルゴリズム (CNSA) スイートに準拠する必要があります。2022 年、NSA は CNSA スイートをバージョン 1.0 からバージョン 2.0 に更新しました。CNSA 2.0 の主な目的の一つは、PQC の導入でした。CNSA スイートを新たに導入する場合は、2027 年 1 月 1 日までにバージョン 2.0 に準拠する必要があります。すでに導入済みのシステムは、2030 年 12 月 31 日までに CNSA 2.0 に準拠するか、CNSA 2.0 準拠のシステムに置き換える必要があります。

表 6-1 に、CNSA 1.0 で承認されている暗号プリミティブと、CNSA 2.0 で承認され必須となっている暗号プリミティブの比較を示します。

表 6-1. CNSA 1.0 と CNSA 2.0 で承認されている暗号プリミティブの比較

アプリケーション用途	暗号プリミティブ	CNSA 1.0 認定	CNSA 2.0 認定 [2]
汎用暗号	対称暗号	連邦情報処理標準 (FIPS) 出版物 (PUB) 197 (高度暗号化標準 [AES]-256 のみ)	
	セキュア ハッシュ	FIPS PUB 180-4 (セキュア ハッシュ アルゴリズム [SHA]-384 のみ)	FIPS PUB 180-4 (SHA-384 または SHA-512)
	デジタル署名	FIPS PUB 186-4 (ECDSA P-384 または Rivest、Shamir、Adleman の 3072 ビット モジュラス)	FIPS PUB 204 (モジュール格子ベース デジタル署名標準 [ML-DSA]-87 のみ)
	鍵合意	楕円曲線 Diffie-Hellman P-384 または有限体 Diffie-Hellman (3072 ビット モジュラス)	FIPS PUB 203 (モジュール格子ベース鍵カプセル化メカニズム [ML-KEM]-1024 のみ)
ファームウェアまたはソフトウェアの署名のみ	デジタル署名	-	NIST 特別出版物 (SP) 800-208 (Leighton-Micali Signature [LMS] または拡張 Merkle Signature Scheme [XMSS][1])
ハードウェアの完全性チェックのみ	セキュア ハッシュ	-	FIPS PUB 202 (SHA-3-384 または SHA-3-512)

7 対称暗号化とハッシュ関数

CNSA 2.0 では、対称暗号プリミティブとセキュア ハッシュ関数の最小要件は大きく変わりません。たとえば、CNSA 2.0 では、対称暗号に CNSA 1.0 と同じアルゴリズムとセキュリティ レベル (AES-256) が引き継がれています。これは、現時点で CRQC が AES-256 に対して大きな脅威とならないためです。AES のセキュリティを低下させる量子アルゴリズムは

Copyright © 2026 Texas Instruments Incorporated

存在しますが、CNSA 1.0 と 2.0 で選定されている鍵サイズ (256 ビット) は十分に大きいため、このプリミティブの安全性は保たれます。例として、Grover のアルゴリズムを考えてみます。Grover のアルゴリズムは、AES に対する総当たり攻撃の時間を 2k から 2k/2 に短縮できる量子アルゴリズムです。ここで、k はビット単位の鍵サイズです。Grover のアルゴリズムを使った場合、AES-256 のセキュリティ強度は、最良の場合でも 2^{256} から 2^{128} に低下するだけで、これは依然として安全と考えられています。一方、AES-128 を使用する設計では、セキュリティ強度が 2^{128} から 2^{64} に低下し、脆弱性が存在します。そのため、現在 AES-128 を使っているアプリケーションでは、AES-256 に切り替えることを推奨します。これにより、量子コンピュータによる優位性を無効にできます。

CNSA 2.0 の最も大きな変更点のいくつかは、デジタル署名と鍵合意に関するものです。

8 PQC デジタル署名:ML-DSA

CNSA 2.0 では、ファームウェアやソフトウェアの更新に使う署名を含め、汎用のデジタル署名に ML-DSA を使うことが求められています。ML-DSA の安全性はモジュール問題の困難性に基づいています。古典コンピューティングと量子コンピューティングの両方の能力を持つ攻撃者に対しても安全であると考えられています。ML-DSA は、NIST に提出された Crystals-Dilithium 署名方式を基に設計されています。NIST は三回の分析と評価を経て、この方式を FIPS 204 として標準化することを決定しました [4]。

デジタル署名に ECDSA を使う組み込みアプリケーション (セキュア ブート、アテステーション、その他の署名用途を含む) に対して、ML-DSA は同様のプログラミング インターフェイスを備えた耐量子安全な代替手段となります。また、ML-DSA は拡張性にも優れ、三つのセキュリティ レベル (ML-DSA-44、ML-DSA-65、ML-DSA-87) をサポートしています。各レベルには、鍵と署名のサイズ、演算レイテンシ、演算に必要なスタティック ランダム アクセス メモリ (SRAM) 容量、セキュリティの間にそれぞれトレードオフがあります。CNSA 2.0 では、最高のセキュリティ レベルである ML-DSA-87 が義務付けられています。

組込みシステムの開発者にとって、ML-DSA は次の点で ECDSA と大きく異なります:

- ML-DSA は、鍵と署名のサイズが ECDSA よりはるかに大きくなります。ML-DSA-87 の署名サイズ (4627 バイト) は、ECDSA P-384 (96 バイト) の 48 倍以上です。そのため、アテステーション機能の実行時や、新しいソフトウェアやファームウェアの更新のインストール時に、フラッシュと SRAM にそれだけ多くの容量を確保する必要があり、通信路での送信時間も長くなります。
- CPU ベースの実装で比較すると、ML-DSA の検証は ECDSA の検証より高速な傾向があり、認証時間を短縮できます。たとえば、ML-DSA-87 の署名検証処理は、NIST P-384 曲線を使う ECDSA の 10 分の 1 の時間で実行できます。そのため、マイコンのブート時間が大幅に短縮されます。
- ML-DSA の実装では一般に、必要な演算を実行するのに ECDSA より多くの SRAM が必要になります。また、作業用メモリと実行レイテンシの間のトレードオフは、セキュリティ レベルによって異なります。

表 8-1 に、ML-DSA の各セキュリティ レベルについて、古典コンピューティングの署名 (ECDSA P-384) と耐量子コンピューティングの署名 (ML-DSA) の比較を示します。

表 8-1. ECDSA と ML-DSA のパラメータ サイズおよび性能の比較

パラメータ	ECDSA	ML-DSA-44	ML-DSA-65	ML-DSA-87
セキュリティ レベル	P-384	カテゴリ 2	カテゴリ 3	カテゴリ 5
秘密鍵 (バイト)	48	2,570	4,032	4,896
公開鍵 (バイト)	96	1,312	1,952	2,592
署名 (バイト)	96	2,420	3,309	4,627
署名の送信時間 (秒)(1)	0.009	0.22	0.30	0.42
鍵生成時間 (秒)(2)	1.12	0.084	0.14	0.23
署名生成時間 (秒)(2)	1.23	≅0.4	≅0.8	≅1.6
検証時間 (秒)(2)	**2.26**	**0.10**	**0.16**	**0.26**
検証用の SRAM 要件 (キロバイト)(2)	2.2	21.9	29.1	37.6

(1) 100kHz、1 バイトあたり 9 ビットでの署名の送信時間
(2) テスト環境は、TI の MSP 暗号化ライブラリを実行するテキサス インスツルメンツ (TI) の 80MHz Arm® Cortex®-M0+ マイコンです

Copyright © 2026 Texas Instruments Incorporated

9 ハードウェア アクセラレーションは必要ですか？

楕円曲線暗号を使う組み込みシステムの開発者は、アプリケーションでレイテンシが重要な場合、楕円曲線演算のアクセラレーションを備えたマイコンを使うことがあります。ただし、その結果、アプリケーションで実際に必要なものより高価なマイコンを選んでしまう可能性があります。

たとえばチャレンジ レスポンス方式のアプリケーションでは、80MHz の Arm Cortex-M0+ プロセッサで ECDSA P-384 署名を計算するのに 2 秒以上かかることがあります。これは、生成された署名を 100Kbps で通信線に送信するのにかかる 6 ミリ秒よりはるかに長い時間です (100Kbps は、スタンダード モードの I^2C や 115200 のシリアル汎用非同期送受信器などの通信方式を想定した一般的な速度です)。

PQC に移行すると、デジタル署名が大きくなり、計算時間が短くなるため、この関係は大きく変わります。たとえば、署名の送信に必要な時間は 48 倍に増え、約 9ms (ECDSA P-384) から約 420ms (ML-DSA-87) になります。署名検証の計算時間は約 2200ms (ECDSA P-384) から約 260ms に短縮されます。これは、署名の送信にかかる時間の半分近くです。その結果、アテステーションなどのアプリケーションでは、計算時間より通信時間がボトルネックになる傾向があり、CPU ベースの実装でも経済的に対応できます。図 9-1 に、ターゲットの Arm Cortex-M0+ マイコンと Arm Cortex-M33 マイコンにおける、セキュリティ レベルごとの検証性能の比較を示します。

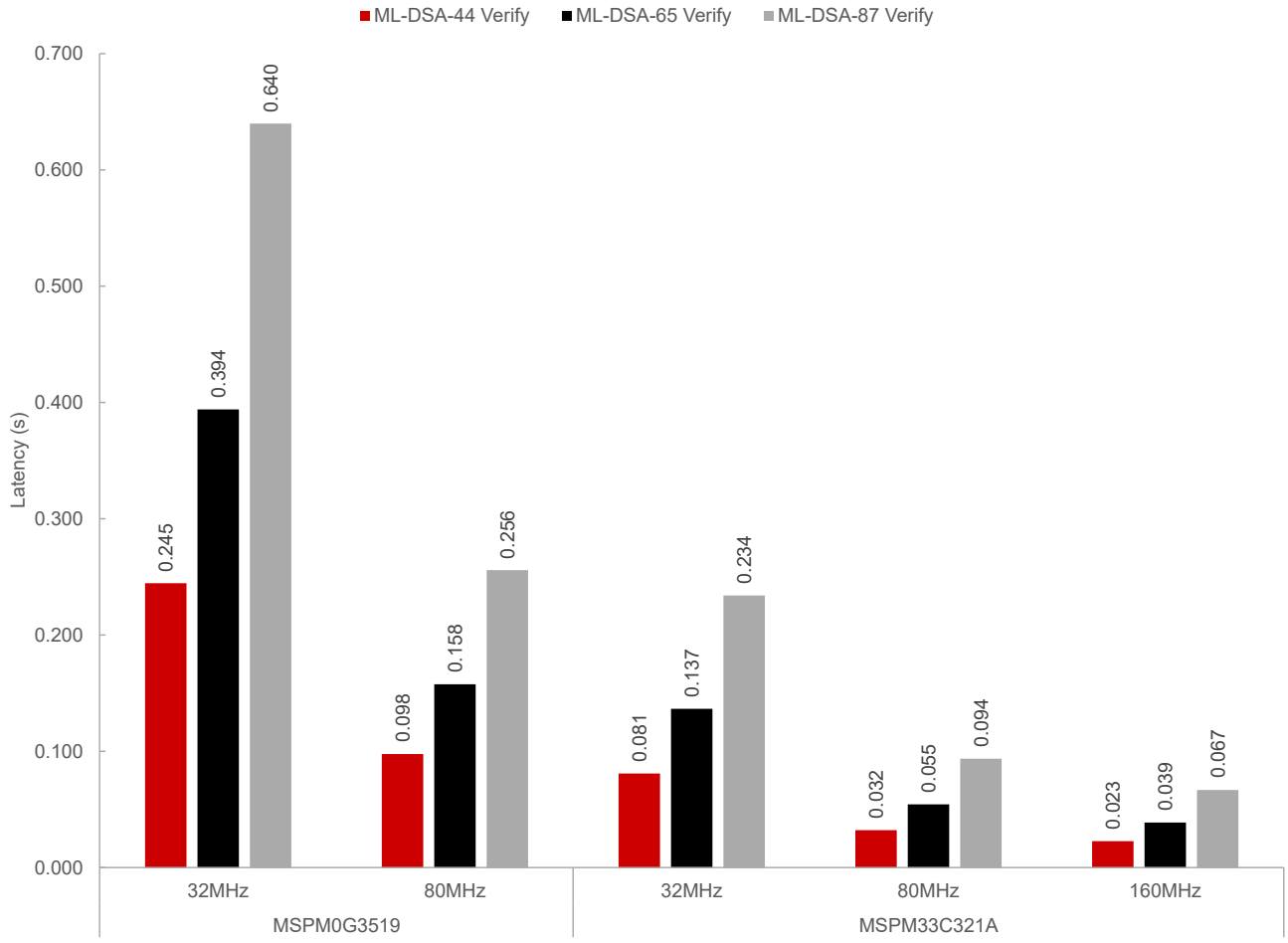


図 9-1. TI MSPM0 および MSPM33 マイコンにおける ML-DSA の検証性能

Copyright © 2026 Texas Instruments Incorporated

10 プリハッシュは必要ですか？

セキュア ブートの実装では、まず SHA-256 や SHA-384 などのセキュア ハッシュ関数を使ってアプリケーション ソフトウェアとファームウェア イメージをハッシュし、固定サイズのメッセージ代表値を作成するのが一般的です。その値に ECDSA で署名し、検証します。ML-DSA では、この「プリハッシュ」ステップが不要です。実際、ML-DSA の設計者は、デジタル署名の対象となるメッセージの内容全体を署名アルゴリズムに直接渡すことを推奨しています。

FIPS PUB 204 では、このプロセスを「pure」モードと呼んでいます。FIPS PUB 204 には HashML-DSA というプリハッシュ モードも定義されています。このモードでは、署名の生成に使ったハッシュ関数の情報が署名に含まれます (つまり、ハッシュ関数が署名にバンディングされます)。

FIPS 204 の規定とは別に、三番目の方法もあります。メッセージをプリハッシュしたうえで、FIPS 204 のプリハッシュ モードである HashML-DSA を使うのではなく、pure モードの ML-DSA でハッシュ値に署名する方法です。この場合、ハッシュ値はメッセージを ML-DSA に直接渡したものとして扱われます。この方法では、ハッシュ関数と署名のバインディングはなくなりますが、現在 ECDSA を SHA と組み合わせて使っている方法により近くなります。

ML-DSA は、デジタル署名の生成と検証の際に、内部で Keccak 関数を使います。Keccak は、SHA-3 関数と、SHA および Keccak の拡張可能出力関数 (Shake) の内部で使われる中核関数です。ML-DSA は、メッセージの安全なダイジェスト生成や、検証時の公開鍵の決定論的な展開など、さまざまな場面で Keccak を使います。

これはリソースに制約のあるシステムにとって重要です。ML-DSA を組み込むと Keccak も組み込まれるため、同じコード フットプリントの中で安全なハッシュ関数を利用できるからです。そのため、pure モードの ML-DSA を使えば、SHA-384 や SHA-512 などのハッシュ関数を別に格納するために、不揮発性コード メモリを余分に割り当てる必要がありません。さらに、Keccak を利用して、アプリケーション内のほかの用途に使う汎用ハッシュ関数として SHA-3 を提供することもできます。

現時点では、CNSA 2.0 準拠のシステムは、承認済みのセキュア ハッシュ関数 (SHA-384 または SHA-512) を使ってメッセージをプリハッシュできます。ただし、FIPS PUB 204 で規定されているプリハッシュ モード (HashML-DSA) で ML-DSA を使うことはできず、FIPS PUB 204 の pure モードがデフォルトです。

11 NIST が選定したその他の PQC デジタル署名方式

NIST は ML-DSA に加えて、標準化の対象として二つの耐量子デジタル署名方式を選定しました:

- Sphincs+ 署名方式に基づき、FIPS PUB 205 として標準化されたステートレス ハッシュベース デジタル署名標準 (SLH-DSA)。
- N 次切り捨て多項式環ユニット格子に基づく高速フーリエ変換デジタル署名アルゴリズム (FN-DSA)、別名 Falcon (FIPS PUB 206 として標準化が進行中)。

SLH-DSA はハッシュベースの署名方式です。ML-DSA と同様に、ステートレスです。ML-DSA とは異なり、高いセキュリティ レベルでは署名サイズが非常に大きくなります (数十キロバイト)。そのため、組み込みアプリケーション、特にマイコンベースのアプリケーションでは、SLH-DSA の適用範囲が限られます。現時点では、NIST が SLH-DSA を標準化しているにもかかわらず、CNSA 2.0 では SLH-DSA ベースの署名は認められていません。

FN-DSA は、実装が複雑になる代わりに、ML-DSA や SLH-DSA より小さな署名を実現します。FN-DSA は浮動小数点演算に依存しています。浮動小数点演算は、プロセッサ アーキテクチャを問わず定数時間で実装するのが非常に難しいことで知られています。また、すべての組込みアーキテクチャに浮動小数点演算ユニットが搭載されているわけではないため、性能上の利点が失われます。現在、CNSA 2.0 では FN-DSA は認められておらず [3]、NIST もまだ FN-DSA を標準化していません。

これらの理由から、業界では、性能、サイズ、セキュリティのバランスに優れた ML-DSA を事実上の標準のデジタル署名方式として採用する動きが広がっています。

Copyright © 2026 Texas Instruments Incorporated

12 ステートフル デジタル署名方式

ML-DSA と、これまでに説明した方式は、すべてステートレス方式です。ステートレス方式では、ある秘密鍵で署名する際に状態を保持する必要がありません。これは組み込みアプリケーションで求められる特性で、特に、リソースに制約のある環境で通常動作中にチャレンジ レスポンス署名が頻繁に行われるアテステーション用途で重視されます。

ステートレス方式が標準化される前から、耐量子安全なステートフル署名方式がいくつか存在していました。NIST SP 800-208 で標準化された LMS や XMSS などで、ソフトウェアやファームウェア イメージの署名 (セキュア ブート) といった特定の用途で使えるオプションでした。

これらの方式の公開鍵は ML-DSA より小さく、新しいファームウェアを認証する際に必要な組み込み不揮発性メモリ (通常はフラッシュ メモリ) が少なくて済みます。そのため、セキュア ブートを実装する組み込みシステムにとって有利です。ただし、署名処理では、秘密署名鍵の秘匿性を確保するために、その鍵の状態を追跡する必要があります。ファームウェアやソフトウェアの更新であれば、これは許容できる場合があります。ファームウェアやソフトウェア イメージの更新は頻繁ではなく、管理された IT 環境で署名用ハードウェア セキュリティ モジュールの状態を維持できるからです。しかし、組込み環境では、これを許容できないことがよくあります。

そのため、LMS と XMSS は CNSA 2.0 準拠のシステムで使用が認められていますが、用途はファームウェアとソフトウェアのデジタル署名に限られます。アテステーションなど、ほかの用途には使えません。これに対し、ML-DSA は、セキュア ブートやアテステーションを含め、デジタル署名のあらゆる用途に使えます。

13 鍵合意はどうでしょうか？

鍵契約は、安全でないチャネルを介して二つのエンティティ間で共有秘密を確立するために必要なステップです。デジタル署名と同様に、有限体 Diffie-Hellman (FFDH) や楕円曲線 Diffie-Hellman (ECDH) など、現在広く使われているプリミティブに依存する鍵合意アルゴリズムも、Shor のアルゴリズムを使った量子攻撃に対して脆弱です。これに対処するため、NIST は ML-KEM (モジュール格子ベースの鍵カプセル化メカニズム) を FIPS 203 で標準化しました。ML-KEM は耐量子安全性を備えた鍵カプセル化メカニズム (KEM) で、現在 FFDH や ECDH が使われている場面で使用することを想定して設計されています。

ハンドシェイクの両側のランダム性から共有秘密を導出する Diffie-Hellman 方式の設計とは異なり、KEM には鍵生成、カプセル化、カプセル化解除の三つのステップがあります。クライアント サーバー モデルでは、ML-KEM による鍵合意のプロセスは次のとおりです:

1. まず、サーバーが鍵ペアを生成します。カプセル化鍵 (「公開」鍵) をクライアントに配布します。カプセル化解除鍵 (「秘密」鍵) はサーバーが保持します。
2. クライアントは、サーバーとの間でセキュアなチャネルを確立して対称鍵を合意する際に、ランダム シードに基づいて ML-KEM のカプセル化処理を実行し、暗号文と共有秘密を得ます。次に、クライアントは暗号文をサーバーに送信します。
3. サーバーは暗号文を受け取ってカプセル化解除処理を実行し、クライアントと同じ共有秘密を得ます。
4. これで、サーバーとクライアントは共有秘密鍵を使って通信できます。

ML-KEM はソフトウェアでも高い性能を発揮でき、160MHz の Cortex-M33 マイコンではカプセル化時間が最短 7ms です。図 13-1 に、ターゲットの Arm Cortex-M0+ マイコンと Arm Cortex-M33 マイコンにおける、セキュリティ レベルごとのカプセル化性能の比較を示します。

Copyright © 2026 Texas Instruments Incorporated

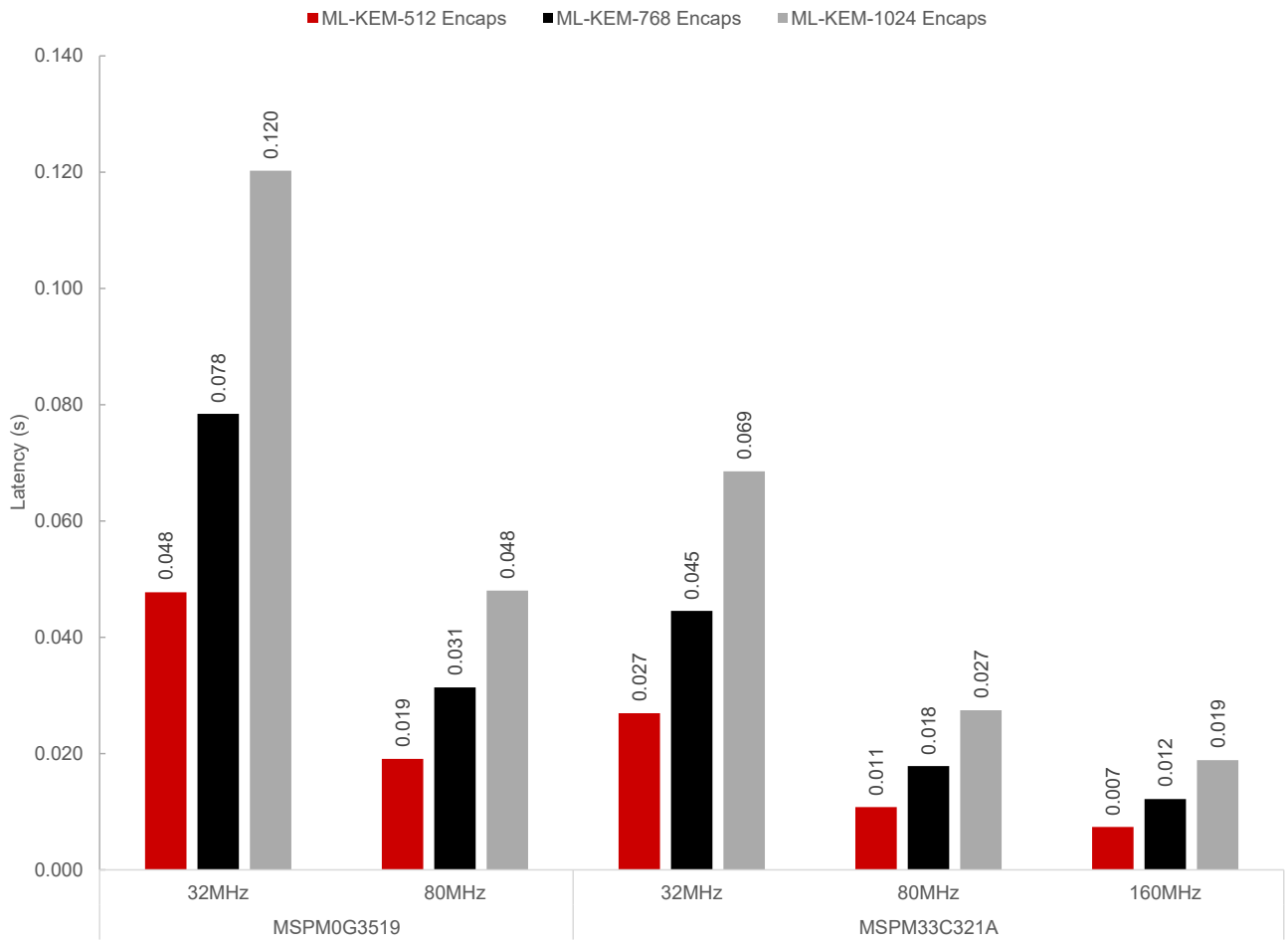


図 13-1. TI MSPM0 および MSPM33 マイコンにおける ML-KEM のカプセル化性能

ML-KEM には、耐量子安全性を備えるように設計されていること以外にも、望ましいセキュリティ特性があります。特に、IND-CCA2 (適応的選択暗号文攻撃に対する識別不可能性) をサポートしています。つまり、クライアントからサーバーへの転送中 (カプセル化からカプセル化解除までの間) に攻撃者が暗号文を改ざんできたとしても、攻撃者が優位に立ったり新たな情報を得たりすることはありません。カプセル化解除の出力は、常にランダムな値と区別できないためです。さらに、ML-KEM の出力は対称暗号鍵としてそのまま使えます。そのため組み込みアプリケーションをシンプルにでき、追加の鍵導出ステップも不要になる場合があります。

14 組込みシステムの将来への備え

設計者は、セキュア ブート、アテステーション、セキュアな通信などの基本的なセキュリティ目標の実現に使われている古典暗号プリミティブを、攻撃者が将来、十分な能力を持つ量子コンピュータで攻撃できるようになる可能性を考慮する必要があります。米国は NIST と NSA を通じて、そのような能力を持つ攻撃者に対しても安全であると考えられる暗号プリミティブを選定するためのフレームワークを整備しています。

組込みシステムの課題には、不揮発性メモリと作業用メモリの容量が小さいこと、通信インターフェイスが低速であること、状態を保存する能力が限られていることなどがあります。TI は、超低消費電力で低コストのマイコンで耐量子安全な暗号プリミティブをサポートしています。これにより、CNSA 2.0 に準拠でき、性能とメモリのバランスも取れます。サポート内容は次のとおりです:

- デジタル署名:ML-DSA-44、ML-DSA-65、ML-DSA-87 をプリハッシュ形式と pure 形式の両方で完全にサポートします。
- 鍵カプセル化:ML-KEM-512、ML-KEM-768、ML-KEM-1024 を完全にサポートします。
- 共通鍵暗号:AES-256 と関連する暗号利用モードに対応しています。これには、暗号ブロック連鎖メッセージ認証コード付きカウンタ モードとガロア / カウンタ モードが含まれます。

Copyright © 2026 Texas Instruments Incorporated

- セキュア ハッシュ関数：SHA-2、SHA-3、SHAKE の各スイートを完全にサポートします。

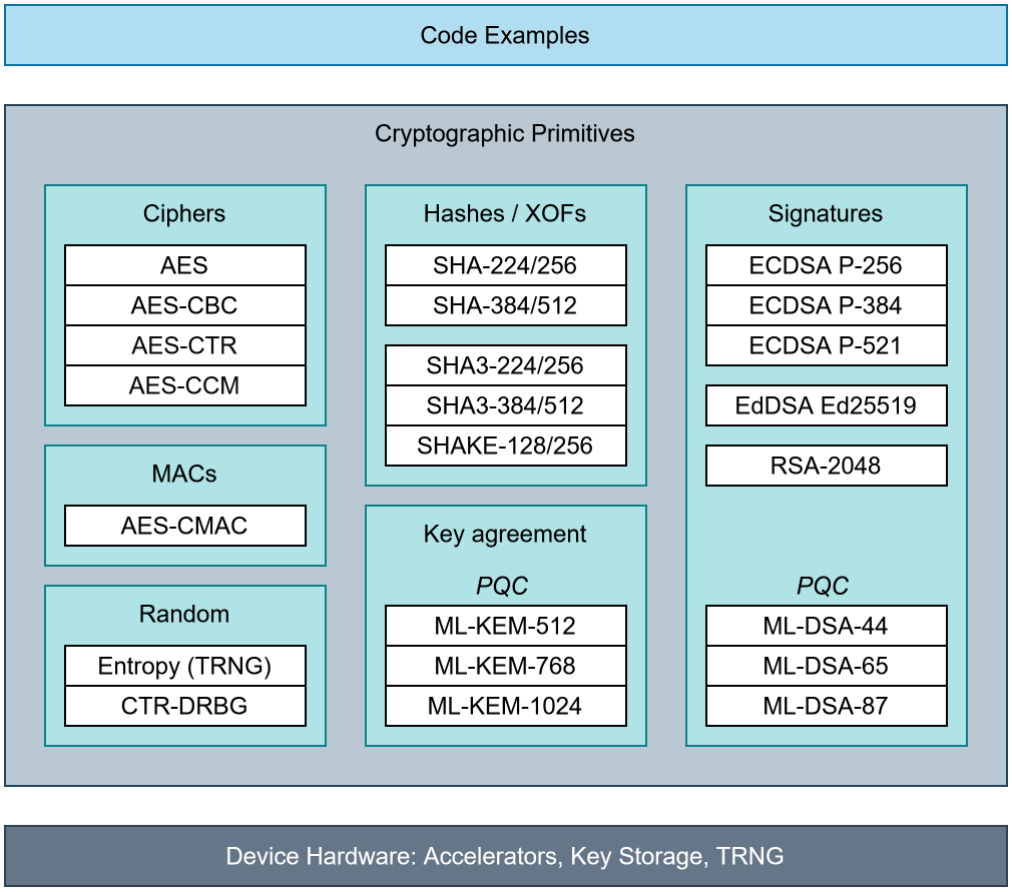


図 14-1. MSP-CRYPTO-LIB の機能

MSPM0 Arm Cortex-M0+ および MSPM33 Arm Cortex-M33 ベースのマイコンで TI MSP Crypto Library を使うと、セキュア ブート用途の CNSA 2.0 準拠 ML-DSA-87 デジタル署名検証を 260ms 未満 (Cortex-M0+) または 70ms 未満 (Cortex-M33) で実行できます。リソースやコストの制約が厳しい環境でもブート時間を短縮でき、量子コンピューティングが現実となる将来にも備えられます。

15 参考資料

1. NIST。2024 年 8 月 13 日。「ポスト量子暗号とは？」アクセス日：2026 年 2 月 3 日。
2. NSA。2022 年 9 月。「Commercial National Security Algorithm Suite 2.0 の発表」アクセス日：2026 年 2 月 3 日。
3. NSA。2024 年 12 月。「The Commercial National Security Algorithm Suite 2.0 と Quantum Computing FAQ」アクセス日：2026 年 2 月 3 日。
4. NIST FIPS 204。2024 年 8 月 13 日。「モジュール - Lattice ベースのデジタル署名標準」アクセス日：2026 年 6 月 6 日

Copyright © 2026 Texas Instruments Incorporated

重要なお知らせと免責事項

TI は、技術データと信頼性データ (データシートを含みます)、設計リソース (リファレンス デザインを含みます)、アプリケーションや設計に関する各種アドバイス、Web ツール、安全性情報、その他のリソースを、欠陥が存在する可能性のある「現状のまま」提供しており、商品性および特定目的に対する適合性の黙示保証、第三者の知的財産権の非侵害保証を含むいかなる保証も、明示的または黙示的にかかわらず拒否します。

これらのリソースは、TI 製品を使用する設計の経験を積んだ開発者への提供を意図したものです。(1) お客様のアプリケーションに適した TI 製品の選定、(2) お客様のアプリケーションの設計、検証、試験、(3) お客様のアプリケーションに該当する各種規格や、その他のあらゆる安全性、セキュリティ、規制、または他の要件への確実な適合に関する責任を、お客様のみが単独で負うものとします。

上記の各種リソースは、予告なく変更される可能性があります。これらのリソースは、リソースで説明されている TI 製品を使用するアプリケーションの開発の目的でのみ、TI はその使用をお客様に許諾します。これらのリソースに関して、他の目的で複製することや掲載することは禁止されています。TI や第三者の知的財産権のライセンスが付与されている訳ではありません。お客様は、これらのリソースを自身で使用した結果発生するあらゆる申し立て、損害、費用、損失、責任について、TI およびその代理人を完全に補償するものとし、TI は一切の責任を拒否します。

TI の製品は、 TI の販売条件 、 TI の総合的な品質ガイドライン 、 ti.com または TI 製品などに関連して提供される他の適用条件に従い提供されます。TI がこれらのリソースを提供することは、適用される TI の保証または他の保証の放棄の拡大や変更を意味するものではありません。 TI がカスタム、またはカスタマー仕様として明示的に指定していない限り、TI の製品は標準的なカタログに掲載される汎用機器です。

お客様がいかなる追加条項または代替条項を提案する場合も、TI はそれらに異議を唱え、拒否します。

Copyright © 2026, Texas Instruments Incorporated

最終更新日：2025 年 10 月