WP512 (v1.0) 2019 年 5 月 21 日

Zynq UltraScale+ MPSoC で 暗号化処理を高速化

Zynq® UltraScale+™ MPSoC が内蔵する暗号化アクセラレータを使用することで、ソフトウェアベースのソリューションと比べて暗号化処理能力を 10,000% 以上向上させることが可能です。

概要

Zynq UltraScale+ MPSoC には、AES-GCM-256 ビット、SHA-3/384、および RSA 用の組み込み型暗号化コア [参照 1] や ArmCortex®-A53 プロセッサの Arm® v8 暗号化拡張機能 [参照 2] があるため、暗号化性能を高速化できます。

内蔵する暗号化アクセラレータの性能について、ここでは RTOS (リアルタイム オペレーティング システム) 上および Linux 上のユーザー空間という 2 つのソフトウェア アーキテクチャを条件に説明しています。各シナリオでは、暗号化アクセラレータの性能をソフトウェアベースの暗号化ソリューションと比較しています。これらの性能測定値に基づいて、システム設計者は要件に応じて暗号化の性能を最大限に引き出すことができます。

はじめに

このホワイトペーパーでは、次について説明しています。

- Zynq UltraScale + MPSoC のコンフィギュレーション セキュリティ ユニット (CSU) に含まれている暗号化アクセラレータを利用した場合に Arm Cortex-A53 プロセッサ上で実行されるソフトウェアの性能
- Arm v8 暗号化拡張機能 (Arm v8 Cryptographic Extensions) を利用した場合に Arm Cortex-A53 プロセッサ上で実行される同等のソフトウェア アルゴリズムの性能

これらの両方の結果を、ネイティブの Arm Cortex-A53 ソフトウェア ソリューションと比較しています。

Zynq UltraScale+ MPSoC CSU のにあるアクセラレータには、AES-GCM の 256 ビット暗号化/復号化、SHA-3/384 ハッシュ関数、および RSA の公開/非公開鍵 (最大の鍵サイズは 4096 ビット) 動作用の組み込み型暗号化コアが含まれています。CSU のデフォルト クロックは、Zynq UltraScale+ MPSoC の内蔵オシレーターの 180MHz [参照 1] で動作しますが、テストではクロックを変更し、低電力ドメインの PLL (位相ロック ループ) から外れて動作し、375MHz に設定してより高い性能を達成します。Arm v8 コアの暗号化拡張機能は、AES、SHA-1、SHA-2、および CRC-32 動作の高速化には対応できますが、RSA または SHA-3 動作には対応できません。

全テストの性能測定は、Zynq UltraScale+ MPSoC の Arm Cortex-A53 プロセッサで実行され、外部の DDR (デュアル データ レート) メモリを使用する wolfSSL[参照 3] 組み込みベンチマーキング ソフトウェア バージョン 3.12.0 を利用しました。WolfSSL によって、暗号化機能用のさまざまなハードウェアを利用して暗号化性能を測定できます。Zynq UltraScale+ MPSoC では、組み込み型暗号化コア、Arm v8 暗号化拡張機能、およびソフトウェアベースのアルゴリズムを利用できます。AES-GCM 256 ビットおよび SHA3/384 暗号化アルゴリズムの性能測定には 16 バイトから最大 15,888 バイトまでのデータセットを使用し、RSA の性能測定には RSA-2048 および RSA-4096 を使用しました。

FreeRTOS 上で wolfSSL を実行

Zynq UltraScale+ MPSoC をサポートするザイリンクス SDK 2017.1 上の FreeRTOS に WolfSSL バージョン 3.12.0 を移植して、AES-GCM 256 ビット、SHA3/384、RSA-2048、および RSA-4096 アルゴリズムに対するソフトウェアベース ソリューション、Arm v8 暗号化拡張機能、および Zynq UltraScale+ MPSoC の組み込み型暗号化コアの性能をベンチマーク評価しました。組み込み型暗号化の性能を最大化するため、wolfSSL FreeRTOS ポート用の XilSecure ライブラリ [参照 4] で CSU の DMA キャッシュとエンディアン バイト スワップを無効に設定しました。AES-GCM 256 ビットおよび SHA3/384 アルゴリズムのスループットを測定するため、データ ブロック サイズは 16 バイト、528 バイト、1,024 バイト、4,112 バイト、7,696 バイト、および 15,888 バイトを使用しました。各動作に必要な平均時間を測定するため、RSA-2048 および RSA-4096 の公開/非公開鍵暗号化を実行しました。RSA 動作に、あらかじめ計算された指数値は使用していません。

結果

表 1 は、FreeRTOS 上で wolfSSL ベンチマークを実行した場合の AES-GCM 256 ビット暗号化の結果を示しています。この表は、最も高速で実行された暗号化機能を**太字**でハイライトしており、データ ブロック サイズが 1,024 バイト以下では Arm v8 暗号化拡張機能の方が Zynq UltraScale+ MPSoC の組み込み型暗号化コアよりも優れていることを示しています。一方、データ ブロック サイズが 1,024 バイトを超えると、MPSoC の暗号化コアの方が Arm v8 暗号化拡張機能よりも優れていることを示しています。

表 1: FreeRTOS 動作のスループット、AES-GCM 暗号化

ブロック サイズ	FreeRTOS 動作のスループット: AES-GCM 256 ビット暗号化 (MB/s)		
	ソフトウェア ベース	Arm v8 暗号化拡張機能	Zynq UltraScale+ MPSoC 暗号化コア
16	3.785	**53.041**	4.012
528	7.52	**278.156**	122.285
1,024	7.634	**297.949**	219.531
4,112	7.73	315.976	**555.58**
7,696	7.743	318.9	**691.379**
15,888	7.746	318.949	**844.344**

表 2 は、FreeRTOS 上で wolfSSL ベンチマークを実行した場合の AES-GCM 256 ビット復号化の結果を示しています。この表は、最も高速で実行された暗号化機能を**太字**でハイライトしており、データ ブロック サイズが 1,024 バイト以下では Arm v8 暗号化拡張機能の方が Zynq UltraScale+ MPSoC の組み込み型暗号化コアよりも優れていることを示しています。一方、データ ブロック サイズが 1,024 バイトを超えると、MPSoC の暗号化コアの方が Arm v8 暗号化拡張機能よりも優れていることを示しています。

表 2: FreeRTOS 動作のスループット、AES-GCM 復号化

ブロック サイズ	FreeRTOS 動作のスループット: AES-GCM 256 ビット復号化 (MB/s)		
	ソフトウェア ベース	Arm v8 暗号化拡張機能	Zynq UltraScale+ MPSoC 暗号化コア
16	3.756	**43.414**	2.069
528	7.515	**166.307**	59.153
1,024	7.634	**175.415**	100.244
4,112	7.73	183.429	**237.055**
7,696	7.75	184.771	**291.194**
15,888	7.746	184.854	**373.363**

表 3 では、SHA3/384 の性能は、データ ブロック サイズが 16 バイトの場合のみ、ソフトウェアベースのソリューションの方が Zynq UltraScale+ MPSoC の組み込み型暗号化コアより優れていることがわかります。

表 3: FreeRTOS 動作のスループット、SHA3/384

ブロック サイズ	FreeRTOS 動作のスループット : SHA3/384 (MB/s)	
	ソフトウェア ベース	Zynq UltraScale+ MPSoC 暗号化コア
16	**38.502**	17.934
528	55.087	**328.623**
1,024	56.152	**417.92**
4,112	57.295	**592.763**
7,696	57.615	**628.626**
15,888	57.52	**654.565**

表 4 では、すべての条件で Zynq UltraScale+ MPSoC の組み込み型暗号化コアの方がソフトウェアベースの RSA ソリューションより優れていることがわかります。

表 4: FreeRTOS の平均 RSA 動作時間

RSA 動作	FreeRTOS の平均 RSA 動作時間: wolfSSL v3.12.0 (ms)	
	ソフトウェア ベース	Zynq UltraScale+ MPSoC 暗号化コア
2048 ビットの公開鍵暗号	4.874	**0.552**
2048 ビットの非公開鍵暗号	89.25	**12.846**
4096 ビットの公開鍵暗号	18.519	**1.95**
4096 ビットの非公開鍵暗号	619.47	**95.9**

Linux 上で wolfSSL を実行

Zynq UltraScale+ MPSoC 用ザイリンクス リリース 2018.3 [参照 5] で利用できる Linux 4.14 上で wolfSSL バージョン 3.12.0 を実行しました。これを使用して、AES-GCM 256 ビット、SHA3/384、RSA-2048、および RSA-4096 アルゴリズムに対するソフトウェアベース ソリューション、Arm v8 暗号化拡張機能、および Zynq UltraScale+ MPSoC の組み込み型暗号化コアの性能をベンチマーク評価しました。メモリ管理や実行権限が違うため、Linux で動作するアプリケーションは RTOS アプリケーションとは異なります。したがって、FreeRTOS ポートに適用した変更は、Linux ポートに適用していません。ただし、FreeRTOS ベンチマークで使用した同じデータセットを Linux でも使用しています。

Linux の場合、ユーザー空間のアプリケーションは Armv8 アーキテクチャ上で最も低い例外レベル (EL0) で実行する必要があります。Zynq UltraScale+ MPSoC の暗号化拡張機能は、ユーザー空間アプリケーションから SMC (セキュア モニター コール) を介してのみアクセス可能です。つまり、ユーザー空間アプリケーションは PMU (プラットフォーム管理ユニット) にアクセスし、その後、XilSecure ライブラリ [参照 4] を使用して CSU 内の暗号動作にアクセスします。暗号化処理が完了すると、Linux アプリケーションは CSU からの応答、PMU コールからの応答、SMC からの応答を受信した後に動作を継続します。この方法による高いオーバーヘッドが Linux のベンチマーク結果には反映されています。

結果

表 5 は、Linux 上で wolfSSL ベンチマークを実行した場合の AES-GCM 256 ビット暗号化の結果を示しています。この表は、最も高速で実行された暗号化機能を**太字**でハイライトしており、すべてのデータ ブロック サイズで Arm v8 暗号化拡張機能の方が Zynq UltraScale+ MPSoC の組み込み型暗号化コアよりも優れていることを示しています。これは、Arm v8 暗号化拡張機能は、ユーザー空間から CSU へ直接アクセスでき、大規模なソフトウェア スタックを通過する必要がないためです。

表 5: Linux 動作のスループット、AES-GCM 暗号化

	Linux 4.14 動作のスループット: AES-GCM 256 ビット暗号化 (MB/s)		
ブロック サイズ	ソフトウェア ベース	Arm v8 暗号化拡張機能	Zynq UltraScale+ MPSoC 暗号化コア
16	4.426	**54.309**	0.151
528	8.971	**300.263**	4.68
1,024	9.105	**322.934**	8.66
4,112	9.206	**344.002**	27.769
7,696	9.235	**347.552**	42.572
15,888	9.23	**345.605**	59.948

表 6 は、Linux 上で wolfSSL ベンチマークを実行した場合の AES-GCM 256 ビット復号化の結果を示しています。この表は、最も高速で実行された暗号化機能を**太字**でハイライトしており、すべてのデータ ブロック サイズで Arm v8 暗号化拡張機能の方が Zynq UltraScale+ MPSoC の組み込み型暗号化コアよりも優れていることを示しています。これは、前述した暗号化の場合と同じ理由によるものです。

表 6: Linux 動作のスループット、AES-GCM 復号化

	Linux 4.14 動作のスループット: AES-GCM 256 ビット復号化 (MB/s)		
ブロック サイズ	ソフトウェア ベース	Arm v8 暗号化拡張機能	Zynq UltraScale+ MPSoC 暗号化コア
16	4.397	**34.835**	0.146
528	8.955	**179.661**	4.535
1,024	9.1	**192.258**	8.52
4,112	9.22	**203.652**	27.083
7,696	9.231	**205.579**	41.784
15,888	9.213	**204.456**	59.311

表 7 では、SHA3/384 の性能は、データ ブロック サイズが 1,024 バイト以下の場合、ソフトウェアベースのソリューションの方が Zynq UltraScale+ MPSoC の組み込み型暗号化コアより優れていることを示しています。一方、データ ブロック サイズが 4,112 バイト以上の場合、MPSoC の暗号化コアの方がソフトウェアベースのソリューションよりも優れていることを示しています。

表 7: Linux 動作のスループット、SHA3/384

ブロック サイズ	Linux 動作のスループット: SHA3/384 (MB/s)	
	ソフトウェア ベース	Zynq UltraScale+ MPSoC 暗号化コア
16	**41.58**	0.308
528	**59.632**	9.822
1,024	**60.797**	18.602
4,112	62.025	**66.507**
7,696	62.357	**112.26**
15,888	62.291	**179.718**

表 8 は、Linux の呼び出しによるオーバーヘッドが大きくても Zynq UltraScale+ MPSoC の組み込み型暗号化コアの方がすべてのソフトウェアベースの RSA ソリューションより優れていることを示しています。

表 8: Linux の平均 RSA 動作時間

RSA 動作	Linux の平均 RSA 動作時間: wolfSSL v3.12.0 (ms)	
	ソフトウェア ベース	Zynq UltraScale+ MPSoC 暗号化コア
2048 ビットの公開鍵暗号	4.424	**1.342**
2048 ビットの非公開鍵復号	83.512	**25.71**
4096 ビットの公開鍵暗号	16.736	**4.152**
4096 ビットの非公開鍵復号	569.89	**191.778**

まとめ

SHA3/384 を Linux で実行した結果を除いて、Arm v8 暗号化拡張機能および Zynq UltraScale+ MPSoC の組み込み型暗号化コアによる暗号化処理の高速化は、ソフトウェアベースのソリューションよりも優れていることがわかりました。

表 9 に、ソフトウェアベースのソリューションと比較した場合の暗号化アクセラレータの最大性能向上率を示しています。表 9 は、AES および SHA アルゴリズムの最大ブロック サイズ、および RSA アルゴリズムのすべての鍵サイズと動作を対象とし、Zynq UltraScale+ MPSoC の組み込み型暗号化コアを使用した場合の wolfSSL FreeRTOS ベンチマーク結果に基づく最大性能向上率を示しています。

表 9: 暗号化アルゴリズムの性能向上率 (ソフトウェアベースのソリューションと比較)

アルゴリズム	性能向上率
AES-GCM 256 ビット暗号化	10,800.39%
AES-GCM 256 ビット復号化	4,720.07%
SHA3/384	1,037.98%
RSA の 2048 ビット公開鍵暗号	782.97%
RSA の 2048 ビット非公開鍵復号	594.77%
RSA の 4096 ビット公開鍵暗号	849.69%
RSA の 4096 ビット非公開鍵復号	545.95%

FreeRTOS と Linux 間でソフトウェアベース ソリューションの性能がわずかに向上しているのは、ソフトウェアベース ソリューションで異なるバージョンのコンパイラが使用されたことによります。FreeRTOS ベンチマーク評価では GCC バージョン 6.2.1 を使用し、Linux ベンチマーク評価では GCC バージョン 7.3.1 を使用しています。コンパイラ バージョンの違いによることを確認するために、Linux 用のソフトウェアベース ソリューションの wolfSSL を GCC バージョン 6.2.1 で再コンパイルし、AES-GCM 256 ビットアルゴリズムを再評価しました。その結果、AES-GCM 256 ビットのソフトウェアベース ソリューションの暗号化性能は 7.023MB/s に低下し、復号化性能は 7.022Mb/s に低下しました。このホワイトペーパーでは暗号化アクセラレータの使用に焦点を当てていますが、コンパイラ バージョンの違いによって性能差が生じることから、ソフトウェアの性能を最大化するためには最新ツールを使用することが非常に重要であることがわかります。

図 1 では、このホワイトペーパーで言及したさまざまなベンチマーク シナリオで実行された AES-GCM 256 ビット暗号化アルゴリズムのすべてのベンチマーク結果を比較しています。図 1 から、ソフトウェアベースのソリューションは、これらの高速暗号化ソリューションと比較して明らかにスループットが劣っていることがわかります。FreeRTOS 上で wolfSSL ベンチマークを実行した場合、2,000 バイトを超えるブロック サイズでは Zynq UltraScale+ MPSoC のハードウェア実装された暗号化コアの方が Arm v8 暗号化拡張機能より優れています。FreeRTOS のソフトウェアベース ソリューションは、Linux のソフトウェアベース ソリューションと重なっているため、グラフ上では判読が困難です。MPSoC の AES-GCM 256 ビット暗号化エンジンを Linux から呼び出す際のオーバーヘッドが大きいため、最大性能を達成するには、常に Arm v8 暗号化拡張機能を使用する方が賢明です。

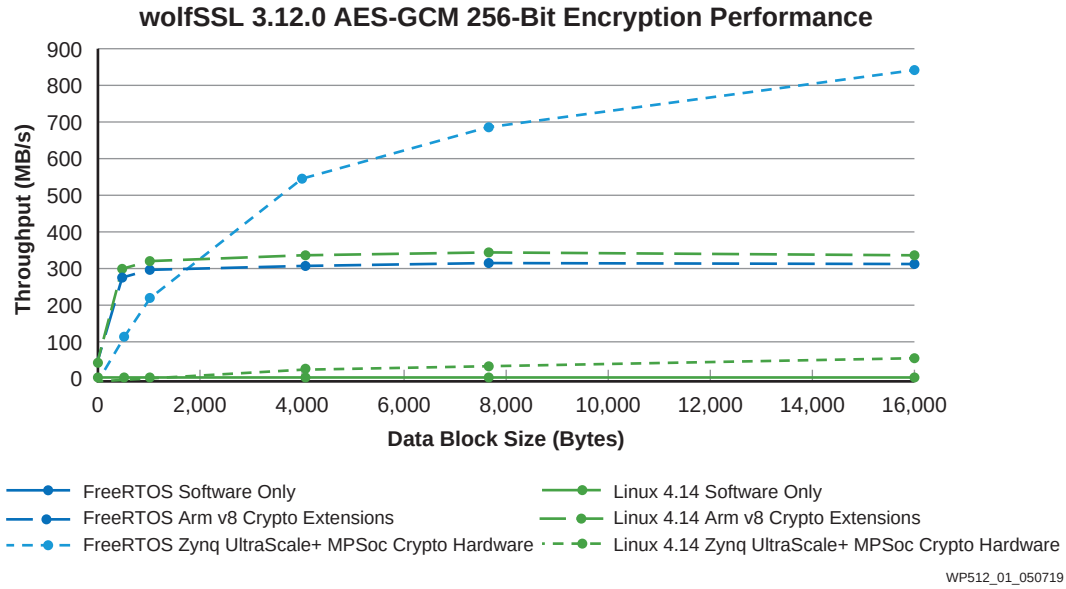

図 1: すべての AES-GCM 256 ビット暗号化の性能比較結果

図 2 では、このホワイトペーパーで言及したさまざまなベンチマーク シナリオで実行された AES-GCM 256 ビット暗号化アルゴリズムのすべてのベンチマーク結果を比較しています。図 2 は図 1 と同様の結果を示していますが、ブロック サイズが 4,000 バイトを超える場合には、FreeRTOS で実行される Zynq UltraScale+ MPSoC の組み込み型暗号化エンジンの性能の方が Arm v8 暗号化拡張機能よりも優れていることを示しています。先ほどと同様、FreeRTOS のソフトウェアベース ソリューションは Linux のソフトウェアベース ソリューションと重なっているため、グラフ上では判読が困難です。

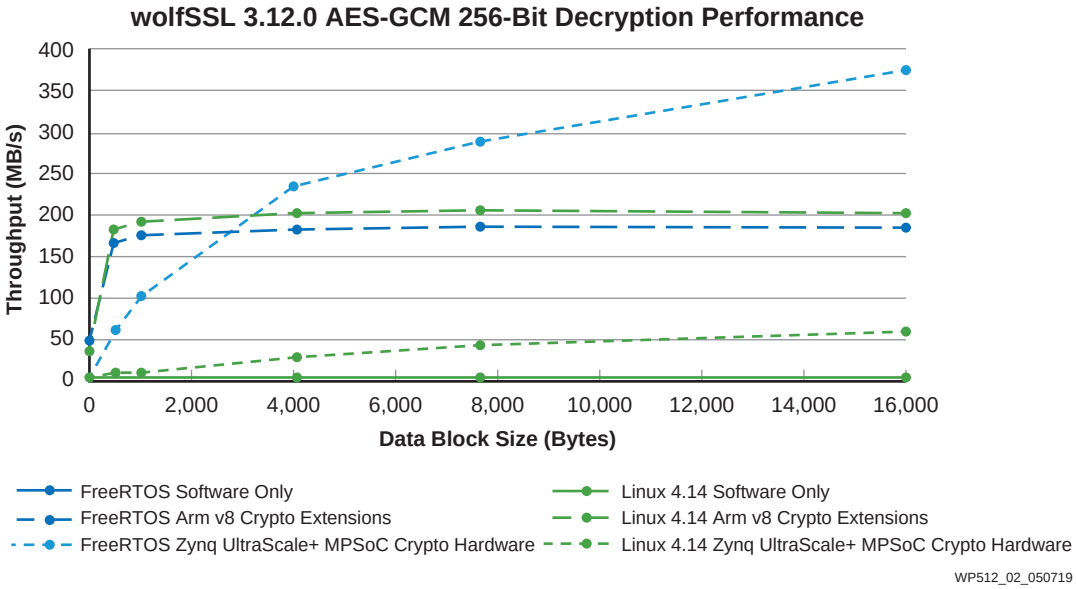

図 2: すべての AES-GCM 256 ビット復号化の性能比較結果

図 3 は、さまざまなベンチマーク シナリオで SHA3/384 アルゴリズムを実行した場合の比較結果を示しています。AES-GCM 256 ビット復号化の結果と同様、FreeRTOS で実行した場合の Zynq UltraScale+ MPSoC の暗号化エンジンの性能は、16 バイトのブロック サイズを使用している場合を除き、その他すべてのソリューションより優れています。呼び出しのオーバーヘッドが大きい Linux でも、4,000 バイトを超えるブロック サイズで Zynq UltraScale+ MPSoC の暗号化アクセラレータを使用すると、ソフトウェアベースのソリューションより優れた性能を達成します。

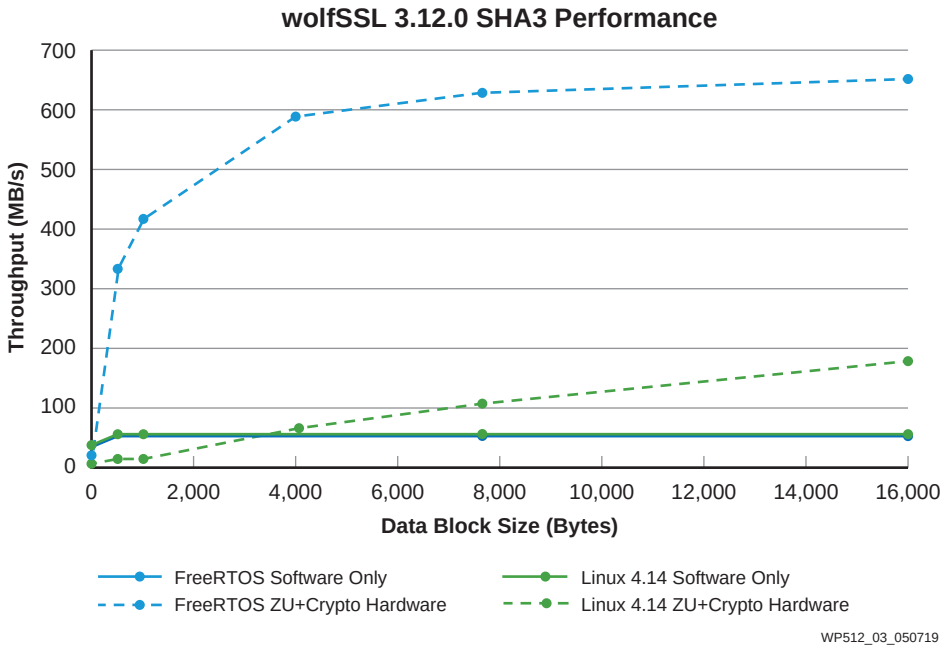

図 3: すべての SHA3/384 の性能比較結果

図 4、5、6、および 7 は、すべての RSA 動作をさまざまなベンチマーク シナリオで実行した場合を総合的に比較しています。これらのグラフから、Zynq UltraScale+ MPSoC の組み込み型暗号化コアを使用した場合の方が、ソフトウェアベース ソリューションよりも常に優れた性能を達成できることがわかります。さらに、Linux に対して FreeRTOS で RSA アルゴリズムを実行して高速化性能を測定した結果、FreeRTOS では CSU に直接アクセスできるため、Linux より常に 2 倍以上の性能向上が可能です。

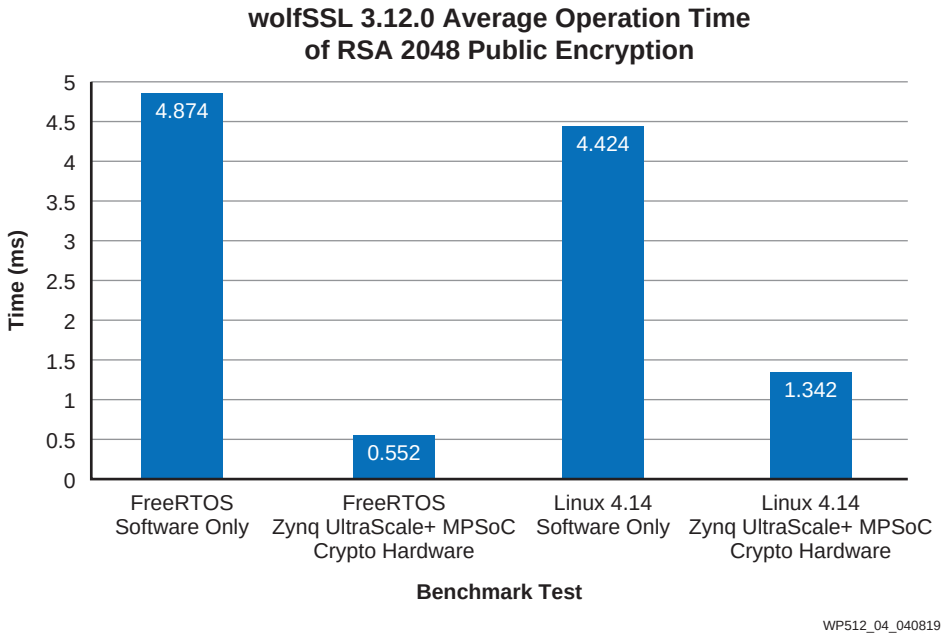

図 4: すべての RSA-2048 公開鍵暗号のベンチマーク結果

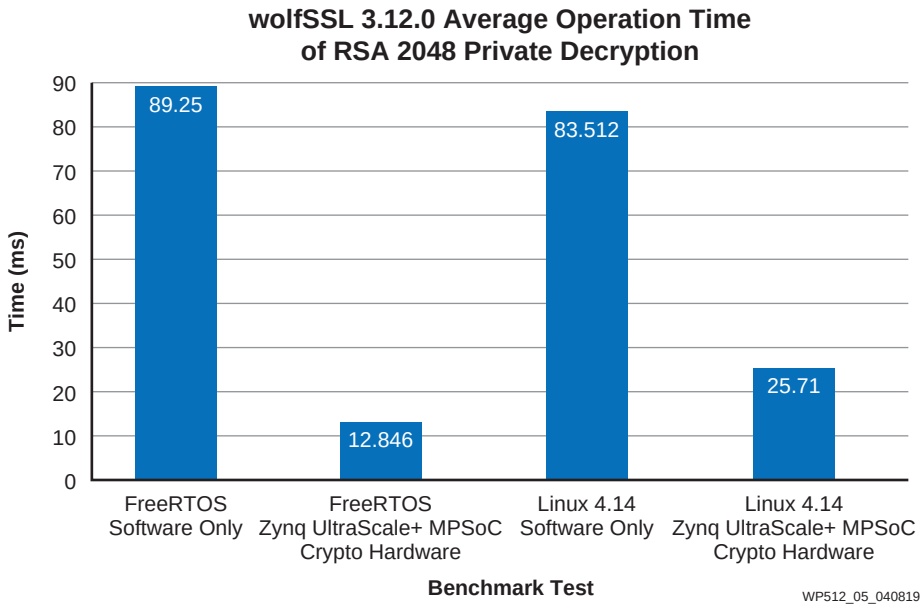

図 5: すべての RSA-2048 公開鍵復号のベンチマーク結果

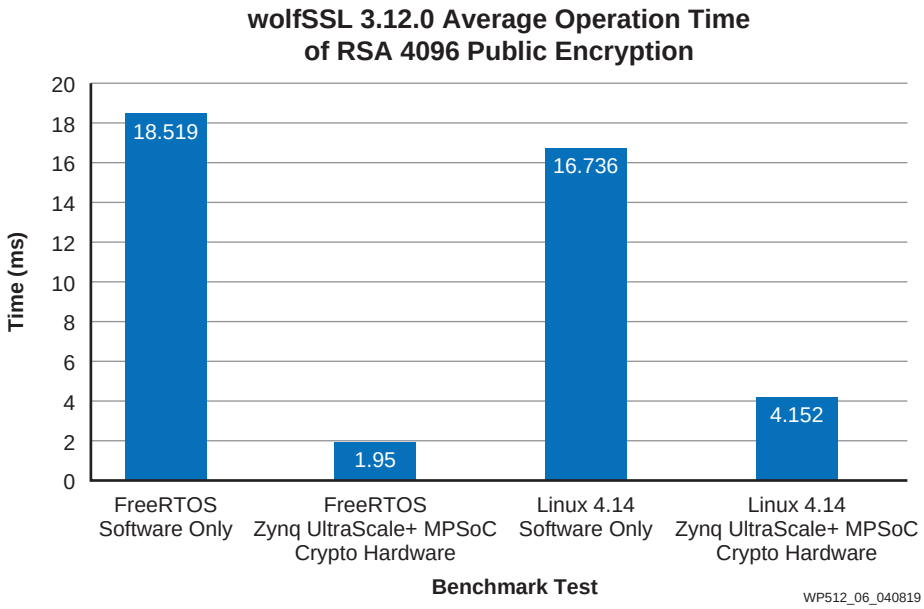

図 6: すべての RSA-4096 公開鍵暗号のベンチマーク結果

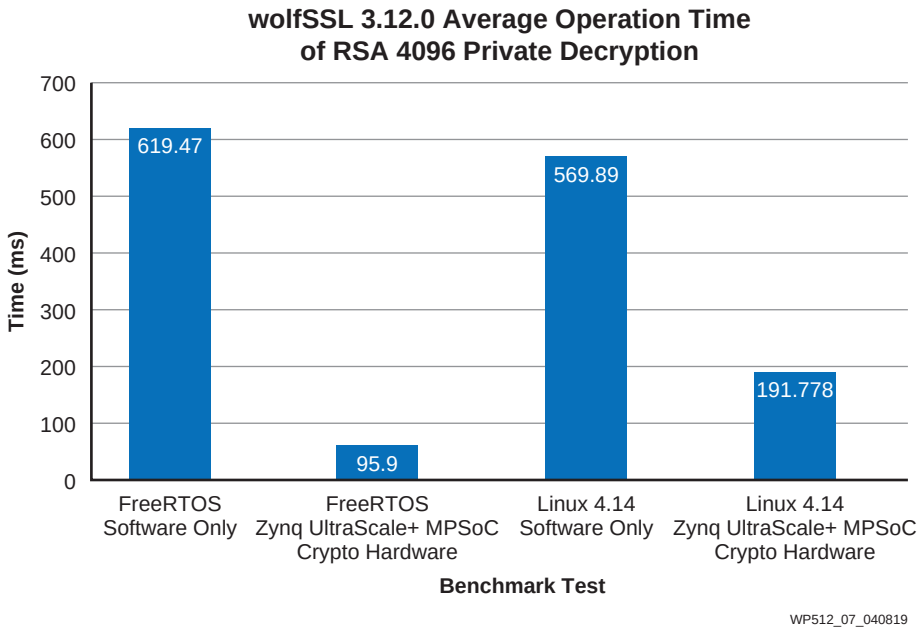

図 7: すべての RSA-4096 公開鍵復号のベンチマーク結果

システム設計者が暗号化ソリューションを定義する際には、次の点に注目して評価する必要があります。

- アルゴリズムの実行方法 - Arm v8 暗号化拡張機能の使用、または Zynq UltraScale+ MPSoC の組み込み型暗号化コアの使用
- 各データ ブロックのサイズ
- アルゴリズムが実行される場所 - シンプルな RTOS、または Linux ユーザー空間
- ソフトウェアが動作するメモリ空間 - DDR、または OCM (オンチップ メモリ)。このホワイトペーパーでは、暗号化アルゴリズムを実行する場所 (DDR と OCM) による性能比較テストは実施していませんが、この点も考慮する必要があります。

このホワイトペーパーで示してきたとおり、これらの変数は Zynq UltraScale+ MPSoC の暗号化性能の高速化に大ききく影響を与えることがあります。このホワイトペーパーでは、考えられるベンチマーク シナリオのごく一部のみを取り上げ、いくつかの有効なソリューションを比較しました。各ユーザーが最終的に自身のデザインをベンチマーク評価して、性能要件が満たされていることを確認する必要があります。

これらの結果から、Zynq UltraScale+ MPSoC 上の暗号化アクセラレータを使用することで暗号化性能を大幅に向上できることを確認でき、AES-GCM-256 ビット アルゴリズム、SHA3/384 アルゴリズム、RSA-2048 アルゴリズム、RSA-4096 アルゴリズムを利用するあらゆる市場、あらゆるアプリケーションの高速化に貢献できることを確信しました。また、これらのアルゴリズムは安全なシステムを実現するための土台を築き、機密性、完全性、認証機能を実現します。最後に、暗号化アクセラレータを使用することで Zynq UltraScale+ MPSoC での性能が向上するだけでなく、Arm Cortex-A53、Arm Cortex-R5、およびプログラマブル ロジックのリソース使用率も軽減できます。このため、Zynq UltraScale+ MPSoC アプリケーションでは、暗号化の要件を満たすためではなく、アプリケーション全体の要件を満たすために、より多くのリソースを確保できます。

参考資料

注記: 日本語版のバージョンは、英語版より古い場合があります。

1. 『Zynq UltraScale+ MPSoC テクニカル リファレンス マニュアル』(UG1085: 英語版、日本語版)
2. Arm テクニカル リファレンス マニュアル: Cortex®-A53 MPCore Processor Cryptography Extension, Revision r0p4
3. wolfSSL ウェブサイト: product landing page
4. 『Zynq UltraScale+ MPSoC ソフトウェア開発者向けガイド』(UG1137: 英語版、日本語版)
5. 『PetaLinux ツール資料: リファレンス ガイド』(UG1144: 英語版、日本語版)

改訂履歴

次の表に、この文書の改訂履歴を示します。

日付	バージョン	内容
2019 年 5 月 21 日	1.0	初版

免責事項

自動車用のアプリケーションの免責条項

この資料に関するフィードバックおよびリンクなどの問題につきましては、jpn_trans_feedback@xilinx.com まで、または各ページの右下にある [フィードバック送信] ボタンをクリックすると表示されるフォームからお知らせください。いただきましたご意見を参考に早急に対応させていただきます。なお、このメール アドレスへのお問い合わせは受け付けておりません。あらかじめご了承ください。