暗号実装における周波数サイドチャネル攻撃のソフトウェア・ガイダンス

この記事は、インテル® デベロッパー・ゾーンに公開されている「Frequency Throttling Side Channel Software Guidance for Cryptography Implementations」の日本語参考訳です。原文は更新される可能性があります。原文と翻訳文の内容が異なる場合は原文を優先してください。

公開日: 2022 年 6 月 16 日

最近のプロセッサーでは、多くの要因によって命令セットの実行時間が変化することがあります。これらの変動要因のうち重要なものは、サイクル数と CPU 周波数です。

インテルは、暗号アルゴリズムを実装する場合、サイクル数の違いによるタイミング・サイドチャネル攻撃を緩和するため、実行時間がデータに依存しない命令 (英語) を選択することを推奨しています。そして、一定時間/一定サイクルのコードを開発する指標として、「Guidelines for Mitigating Timing Side Channels Against Cryptographic Implementations (暗号実装に対するタイミング・サイドチャネル攻撃を緩和するためのガイドライン)」(英語) を提供しています。

この記事では、CPU 周波数に起因するタイミング・サイドチャネル攻撃を緩和するためのソフトウェア・ガイダンスを提供します。インテル® プロセッサーを含むほとんどの最新プロセッサーの電源管理アルゴリズムには、電気パラメーター (電力や電流など) を与えられた制限値以下に維持するメカニズムがあります。これらの制限に達すると、CPU 周波数スロットルがトリガーされ、インテル® ターボ・ブースト・テクノロジーが有効かどうかにかかわらず、CPU 周波数が変化します。この周波数の変動およびそれにかかわる動作は、CPU で処理されている情報と相関する場合があり、周波数の遷移を詳しく分析することで情報の一部を推測できる場合があります。この記事で示すガイダンスは、原文の執筆時点におけるインテルの情報と理解に基づいています。ほかのセキュリティー・ガイダンスと同様に、脅威の状況の変化や新しい情報の入手に伴い、インテルのガイダンスは変更されることがあります。

周波数サイドチャネル攻撃

CPU がデータを処理する際に、処理するデータに応じてトランジスターのスイッチがオン/オフになります。トランジスターのスイッチの切り替えには電力が必要です。そのため、同じワークロードを異なるデータで実行すると、CPU の消費電力が変化する可能性があります。この物理特性により、悪意のある行為者が、システムで処理されている可能性のある秘密のデータと、システムで報告された消費電力を関連付けることが可能性になります。インテルの緩和策の詳細は、インテルの技術記事「Running Average Power Limit Energy Reporting (RAPL 電力レポート)」(英語) を参照してください。

CPU 電源管理ユニットは、過去のタイムウィンドウにおける実行平均電気パラメーターを定期的に計算し、電源管理のリアクティブ制限と比較します。いずれかの制限を超えた場合、電源管理アルゴリズムは CPU スロットルをトリガーし、最大許容周波数を調整します。その結果、平均スロットル周波数と周波数スロットル前の消費電力 [1] の間に**逆相関**が生じます。スロットル前の消費電力が高いワークロードは、平均スロットル周波数が低くなる傾向があり、逆もあります。さらに、ワークロードの消費電力は、処理中のデータと相関する可能性が

あるため、スロットル周波数もデータと相関する可能性があり、周波数サイドチャネル攻撃の対象となる可能性があります。また、CPU 周波数の変化により、ワークロードの実行時間が変化するため、タイミング・サイドチャネル攻撃にもつながる可能性があります。

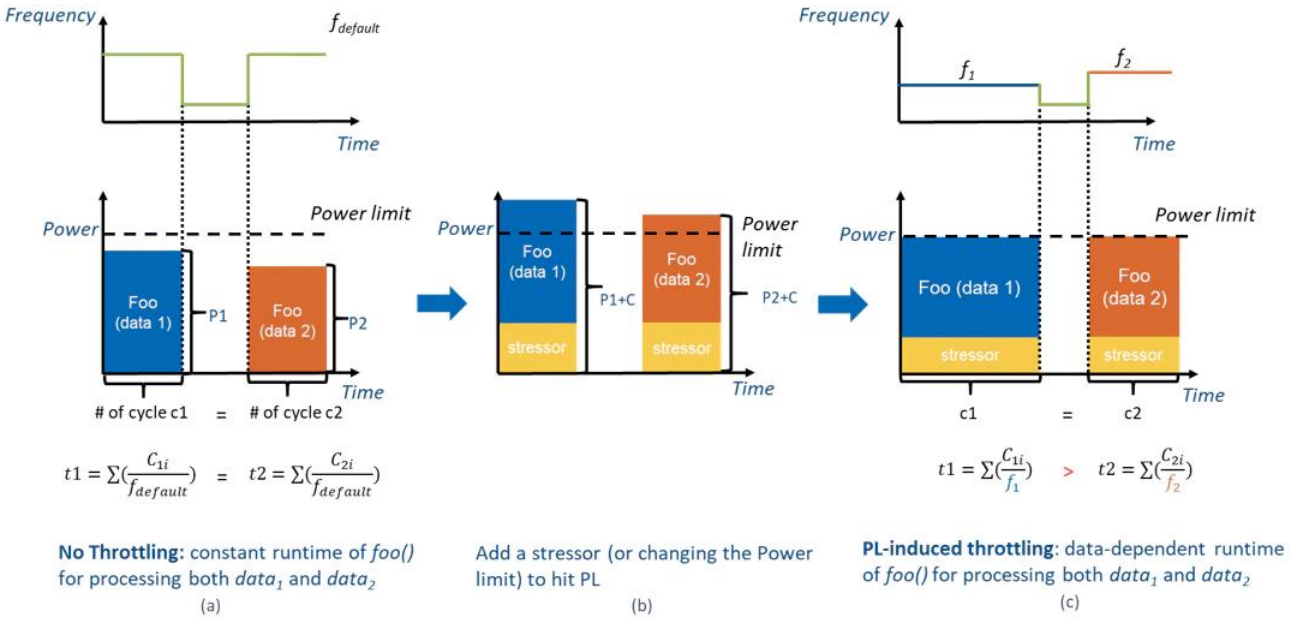

図 1: 電源管理のリアクティブ制限によるスロットルが消費電力の変化を引き起こし周波数/タイミングが変化する

図 1 は、図を使ってサイドチャネル攻撃を解説したものです。図 1 の (a) は、同じプログラムを入力データ 1 (青) と入力データ 2 (オレンジ) で実行したものです。このプログラムが定周期実装の場合、データ 1 とデータ 2 のサイクル数は同じです (c1 = c2)。一方、データ 1 とデータ 2 の処理にかかる消費電力は異なる可能性があります。一般性を損なわない範囲で、データ 1 の処理の消費電力 (p1) ほうがデータ 2 の処理の消費電力 (p2) よりも大きいと仮定します。p1 と p2 のどちらも既定の電力制限値 (またはその他のリアクティブ制限値) を超えていない場合、スロットルは行われず、プログラム実行中の周波数は $f_{default}$ のままとなり、プログラムの実行時間は、データ 1 でもデータ 2 でも同じになります。

システムの消費電力の増加 (高負荷のコードが関数と並列に動作し始めた場合など) や電力制限の引き下げにより、総消費電力が電力制限値を超えると、周波数スロットルが発生します。図 1 の (c) に示すように、データ 1 の処理のほうがデータ 2 の処理よりも消費電力が大きいため、データ 1 の平均スロットル周波数 ($freq_1$) は、電力制限を満たすようにデータ 2 の周波数 ($freq_2$) よりも低くなります。当然、どちらのスロットル周波数も $f_{default}$ より低くなります。したがって、プログラムの実行サイクル数がデータによって変化しない場合あっても、スロットル周波数と実行時間はデータによって変化します。一般的な「定数時間」の暗号実装では、「定周期」の実行しか保証されず、CPU 周波数はデータによって変化するため、コード実行時間もデータ依存となり、攻撃者はこのサイドチャネルを利用して「定数時間」の暗号実装から秘密データ (暗号鍵など) を抽出することが可能なのです。

電源管理のリアクティブ制限

インテル® プロセッサーには、RAPL (Running Average Power Limit) や VR-TDC (Voltage Regulator Thermal Design Current Limit) などの電源管理に関連するいくつかのリアクティブ制限があります。

RAPL (Running Average Power Limit: 実行平均電力制限)

RAPL は、インテルの電源管理アーキテクチャーでサポートされる機能で、システムの消費電力の上限を設定します。設定された電力制限を超えると、CPU は低い周波数での動作を強制され、電力制限の要件を満たしつつパフォーマンスを最大化します。インテルは現在、パッケージレベルの電力制限とプラットフォームレベルの電力制限を含む、複数の電力制限機能を提供しています。Ring 0 ソフトウェアは、MSR_PKG_POWER_LIMIT (パッケージレベルの電力制限) などのモデル固有のレジスター (MSR) を利用して、各機能の実行平均ウィンドウと電力制限の両方を設定できます。詳細は、『インテル® 64 および IA-32 アーキテクチャー・ソフトウェア・デベロッパー・マニュアル第 3 巻』(英語) のセクション 14.10「Platform Specific Power Management Support (プラットフォーム固有の電源管理サポート)」を参照してください。

VR-TDC (Voltage Regulator Thermal Design Current Limit: 電圧レギュレーター熱設計電流制限)

VR-TDC は、インテルの電源管理アーキテクチャーでサポートされている電源管理機能です。この機能は、電圧レギュレーター (VR) の特性に関する電気的制約を維持するため、電流制限 (アンペア数で指定) を導入します。一般に、制御アルゴリズムは、VR から電流測定値を読み取って、同じくアンペア単位で測定される指数移動平均 (EMA) 電流を監視します。ほかの制御アルゴリズムと同様に、このアルゴリズムは、与えられた時間ウィンドウに基づいて電力割り当てを制御します。制限に達すると、プロセッサーは CPU 周波数を下げて (周波数スロットル)、電流がこの制限未満になるように調整します。

関連する問題

インテルは、IPU 2020.2 と IPU 2021.2 で、RAPL 電力レポートの脆弱性 (CVE-2020-8694 および CVE-2020-8695) に対応するマイクロコード・アップデート (MCU) をリリースしています。いずれの MCU も、周波数サイドチャネル攻撃に対する脆弱性を緩和するものではありません。後述の「暗号実装のためのソフトウェア・ガイダンス」セクションでは、暗号実装に対する周波数サイドチャネル攻撃を緩和するソフトウェア・ガイダンスを提供しています。暗号ライブラリーおよび暗号アプリケーションの開発者は、同セクションで提案されている方法を参照して、周波数サイドチャネル攻撃 (別名「Hertzbleed」) についてコードを評価して強化することを推奨します。

暗号実装のためのソフトウェア・ガイダンス

このセクションでは、暗号アプリケーションや暗号ライブラリーの開発者 [2] が、暗号実装の周波数サイドチャネル攻撃のリスクを評価し、その影響を軽減するためのガイダンスを提供します。周波数サイドチャネル攻撃の根本的な原因は消費電力サイドチャネル攻撃であり、その緩和策については徹底的に研究されています。この記事は、すべての暗号実装における周波数サイドチャネル攻撃を緩和する包括的な解決策を提供するものではなく、暗号開発者がリスクを評価し、このサイドチャネル攻撃に対してソフトウェア実装を強化できるように推奨事項を提供するものです。インテルは、暗号実装において、「Guidelines for Mitigating Timing Side Channels Against Cryptographic Implementations (暗号実装に対するタイミング・サイドチャネル攻撃のガイドライン)」(英語) と「Data Operand Independent Timing Instruction Set Architecture Guidance (データオペランドに依存しないタイミング命令セット・アーキテクチャーのガイダンス)」(英語) に記載されている、定周期コードの開発に関する既存のガイダンスに従うことを推奨しています。

攻撃の条件

暗号実装は、以下の条件をすべて満たす場合、周波数サイドチャネル攻撃に対して脆弱となる可能性があります。これらの前提条件の 1 つ以上を満たしていない場合、暗号実装はこのタイプのサイドチャネル攻撃の影響を受けないはずです。

以下の条件リストを確認し、実装の性質と脅威モデルに基づいて、実装のリスクを評価してください。

暗号実装は消費電力サイドチャネル攻撃に弱い

周波数サイドチャネル攻撃の根本的な原因は消費電力サイドチャネル攻撃です。周波数サイドチャネル攻撃に対して脆弱な実装は、電力を測定する物理アクセス機能を除き、物理消費電力サイドチャネル攻撃の前提条件をすべて満たす必要があります。悪意のある行為者が物理的な消費電力サイドチャネルを悪用する前提条件は、以下に限定されるものではありません。

- 十分なデータを収集するため、同じ秘密鍵で繰り返し暗号操作を開始できること。
- ブロック暗号の場合、ブロック暗号プリミティブの入出力やラウンド間の状態を読み取ることができること。ただし、入出力は必ずしも平文や暗号文である必要はありません。例えば、ブロック暗号のカウンター (CTR) モードの場合、ブロック暗号への入力はノンスとカウンターの連結です。

被害者の実行が CPU のリアクティブ制限に達することを保証する

電力周波数と秘密データを相関させるには、被害者がワークロードを実行中にリアクティブ制限に達する必要があります。この必要条件を満たすため、攻撃者はいくつかの手法を取ります。

- 攻撃者は、(同じ秘密データを使用して) 被害者のワークロードの複数のインスタンスを複数のコアで実行し、パッケージの消費電力を増加させ (制限に達するようにし)、SN 比 [3] を増加させます。
- 攻撃者は、パッケージの消費電力を増加させるため、被害者のワークロードと並行して高負荷のワークロードを実行します。
- リング 0 特権を持つ攻撃者は、被害者のワークロードが制限に達するように、リアクティブ制限設定インターフェイス (MSR など) を介して制限を引き下げます。

周波数の変化または関連する CPU 動作を十分な精度で監視する

攻撃者は、被害者がワークロードを実行中に CPU 周波数をサンプリングするか、被害者のワークロードの実行時間を十分な精度で観察して、測定情報からデータ依存の変化を特定します。

ソフトウェア実装

以下のガイダンスは、開発者が周波数サイドチャネル攻撃を緩和するのに役立ちます。このガイダンスは、周波数サイドチャネル攻撃のすべての条件が満たされていない場合でも、多層防御メカニズムとして使用することができます。

消費電力サイドチャネル攻撃に対する効果的なソリューションの適用

暗号プリミティブの消費電力サイドチャネル攻撃に対するソフトウェアベースの対策のほとんどは、周波数サイドチャネル攻撃に対しても効果的です。例えば、単一命令の電力 SN 比を低減する緩和策は、実行平均ウィンドウの平均消費電力 (およびスロットルによるタイミング) の SN 比も低減するため、物理消費電力サイドチャネル攻撃と周波数サイドチャネル攻撃の両方に有効です。ソフトウェアベースのマスキング[1] [2] などのほかの手法も、周波数サイドチャネル攻撃に対して効果があるはずです。

命令の実行順序をランダムにする対策は、物理消費電力サイドチャネル攻撃においてトレース・アライメントと注目点の特定を困難にする効果はありますが、周波数サイドチャネル攻撃の緩和にはあまり効果がありません。これは、サイクル粒度での命令の並べ替えは、ミリ秒以上の平均時間ウィンドウでは、平均消費電力に影響を与える可能性が低いためです。

暗号アプリケーションでは、消費電力サイドチャネル攻撃に対する汎用的な対策の一例として、「鍵の更新」があります。消費電力サイドチャネル攻撃の必要条件の 1 つは、同じ秘密鍵で暗号操作を繰り返し開始できることです。十分なトレースを集める前に秘密鍵が更新されれば、攻撃者が秘密を完全に推論することは難しくなります。鍵の更新頻度は、タイミング (例えば、数時間ごとに更新) やデータ量 (例えば、同じ鍵で暗号化されているデータ量) に基づきます。どのしきい値を使用すべきか分からない場合は、パフォーマンス/設計要件を満たす最も低いしきい値を選択します。鍵の更新の実用性は、暗号のユースケースに依存します (例えば、鍵の更新は通常ディスク暗号化には適用されません)。

リアクティブ制限設定インターフェイスの不要な露出を回避する

「攻撃の条件」で述べたように、攻撃者が特定のハードウェア・インターフェイス (例えば、MSR_PKG_POWER_LIMIT) にアクセスできる場合、被害者のワークロードが周波数スロットルを簡単に起動できるように、リアクティブ制限を変更および削減する可能性があります。攻撃対象領域を減らすため、これらのインターフェイスにアクセスする特権ソフトウェア (ハイパーバイザーやリング 0 ソフトウェアなど) は、これらのインターフェイスを信頼できないエンティティー (ゲスト VM やリング 3 ソフトウェアなど) に不必要に公開することを避けるべきです。これらのインターフェイスを公開する必要がある場合、特権ソフトウェアの設計者は潜在的なセキュリティー上の影響を認識すべきです。

周波数の変化や関連動作の相関を制限する

サイドチャネル攻撃に対するもう 1 つの一般的な対策は、チャネルをノイズで妨害して攻撃者が秘密を推論するのを阻止することです。周波数サイドチャネル攻撃では、周波数の変化または派生する動作が対象となるため、周波数の遷移またはタイミング情報にノイズを加えることができます。

その方法の 1 つとして、暗号アプリケーションの呼び出し時に内在するノイズを活用する方法があります。暗号ライブラリーや暗号アプリケーションの提供者は、同じ量のデータの処理により多くの API 呼び出しが必要となるように、API 呼び出しごとに許容される平文/暗号文の最大サイズを制限できます。これにより、加えるノイズの量がさらに増えます。

このほか、暗号実装者が暗号処理にランダムなノイズを加えて、タイミングのばらつきを大きくすることもできます。この対策を実装するには、十分な消費電力やレイテンシーのばらつきをもたらすダミー命令を追加することが考えられます。ダミー命令は、暗号機能を使用する秘密データから独立したものであるべきです。例えば、ランダムな反復を行うループ命令使用して、タイミングのばらつきを発生させることができます。さらに、ダミー命令によって引き起こされる電力の変化は、周波数遷移のエントロピーを増加させる可能性があります。

すべての周波数変化がノイズの影響を受けるようにするため、インテルは、攻撃者に利用される可能性のあるリアクティブ制限の実行時間ウィンドウに何らかのノイズを加えることを推奨しています。セキュリティーとパフォーマンスのトレードオフのバランスを取る方法として、この方式と「消費電力サイドチャネル攻撃に対する効果的なソリューションの適用」セクションで説明した鍵の更新対策を組み合わせて、攻撃を成功させるのに必要な時間を、実装上許容できる鍵の存続期間よりも長くなるようにすることが考えられます。

コードを保護するために取るべき手順

暗号ライブラリーの提供者は、以下の手順に従って、コードを評価し保護することをお勧めします。

1. 脅威モデルと攻撃の必要条件に基づいて、実装が影響を受けるかどうかを評価します。
2. 暗号実装が影響を受け、緩和が必要な場合は、次の手順を実行します。
    1. 暗号プリミティブ・レベル (例: マスキング) または暗号アプリケーション・レベル (例: 鍵の更新) の消費電力サイドチャネル攻撃に対する汎用的な対策を適用します。
    2. 周波数の変化や関連動作の相関を制限します。例えば、1 回の呼び出しにおける最大入力データサイズを制限したり、ランダムな遅延ノイズを加えます。
3. 特権ソフトウェアやハイパーバイザーでは、リアクティブ制限設定インターフェイスが信頼できないエンティティーに不必要に露出しないようにします。

参考資料

1. S. Mangard, E. Oswald and T. Popp, Power Analysis Attacks: Revealing the Secrets of Smart Cards.
2. E. Prouff and M. Rivain, "Masking against Side-Channel Attacks: a Formal Security Proof," in Annual International Conference on the Theory and Applications of Cryptographic Techniques, 2013.

脚注

1. この記事では、簡略化のため、同様の挙動を示す電気的パラメーターはすべて「電力」と表記しています。
2. 「アプリケーション」とは、暗号ライブラリー・プリミティブを利用し、暗号鍵を所有/管理するソフトウェアです。「ライブラリー」とは、暗号プリミティブを提供し、鍵は所有しないソフトウェアです。暗号ライブラリーは、使用する暗号鍵をアプリケーションから受け取ります。
3. 「信号 (シグナル)」は秘密に関連する消費電力、ノイズは秘密に依存しない消費電力です。
