Google Cloud セキュリティ ホワイトペーパー

2018 年 3 月

Google インフラストラクチャの セキュリティ設計の概要

Google Cloud での 保存時の暗号化

Google Cloud で転送 されるデータの暗号化

Google Cloud での Application Layer Transport Security

Google Cloud

Google Cloud

目次

Google インフラストラクチャのセキュリティ設計の概要

Google Cloud ホワイトペーパー

Google Cloud

Google Cloud

目次

Google Cloud

CIO レベルの概要

- Google には、Google の情報処理ライフサイクル全域を通してセキュリティを確保するように設計されたグローバル規模の技術インフラストラクチャがあります。このインフラストラクチャによって、サービスのデプロイにおけるセキュリティ、データ ストレージのセキュリティ（とエンドユーザーのプライバシー保護）、サービス間の通信のセキュリティ、お客様とインターネット経由の通信の機密性とセキュリティ、管理者オペレーションの安全性が提供されています。

- 検索、Gmail、フォトなどの消費者向けサービスと、G Suite や Google Cloud Platform などの企業向けサービスの両方からなる Google のインターネット サービスは、このインフラストラクチャを使用して構築されています。

- インフラストラクチャのセキュリティは、進行型の階層構造で設計されています。まず、データセンターの物理的なセキュリティがあり、次にインフラストラクチャの基礎となるハードウェアとソフトウェアのセキュリティがあり、最後にオペレーションのセキュリティをサポートする技術的な制限やプロセスがあります。

- Google では、自社のインフラストラクチャと全社に分散した何百人ものセキュリティとプライバシー専門のエンジニア（中には業界の権威として認められている人もいます）に多大な投資を行っています。

はじめに

このドキュメントでは、Google　　の技術インフラストラクチャのセキュリティ設計についての概要を説明します。グローバルな規模を持つこのインフラストラクチャは、Google の情報処理ライフサイクル全域を通してセキュリティを確保するために設計されています。このインフラストラクチャによって、サービスのデプロイにおけるセキュリティ、データ ストレージのセキュリティ（とエンドユーザーのプライバシー保護）、サービス間の通信のセキュリティ、お客様とインターネット経由の通信の機密性とセキュリティ、管理者オペレーションの安全性が提供されています。

検索、Gmail、フォトなどの消費者向けサービスと、G Suite や Google Cloud Platform などの企業向けサービスの両方からなる Google のインターネット サービスは、このインフラストラクチャを使用して構築されています。

このインフラストラクチャのセキュリティについて、進行型の階層構造にそって説明していきます。まず、データセンターの物理的なセキュリティから始まって、インフラストラクチャの基礎となるハードウェアとソフトウェアのセキュリティに続き、最後に、オペレーションのセキュリティをサポートする技術的な制限やプロセスについて説明します。

図 1.
Google インフラストラクチャ セキュリティ レイヤ:

一番下のハードウェア　インフラストラクチャから一番上のオペレーションのセキュリティまでのさまざまなセキュリティのレイヤ。このドキュメントでは、それぞれのレイヤについて詳しく説明します。

下位インフラストラクチャの保護

このセクションでは、インフラストラクチャの一番下のレイヤをどのように保護しているかについて説明します。このレイヤには、物理施設から、データセンター内の専用ハードウェア、すべてのマシン上で動作している下位ソフトウェア スタックまでが含まれます。

物理施設のセキュリティ

Google では、複数の物理的セキュリティ保護レイヤからなるデータセンターを独自に設計して構築しています。これらのデータセンターへのアクセスは、ごく少数の Google 社員に制限されています。複数の物理セキュリティ レイヤを使用してデータセンターのフロアを保護しており、生体認証、金属検知、カメラ、車両障害物、レーザーを使った侵入検知システムなどの技術が利用されています。加えて、一部のサーバーをサードパーティ データセンターでホストしています。ここでも、データセンター オペレーターが提供するセキュリティ レイヤに加え、Google が管理する物理セキュリティ対策が設置されています。たとえば、これらのデータセンターには、独立した生体識別システム、カメラ、金属検知器を配備しています。

ハードウェアの設計と供給元

Google データセンターは、ローカル ネットワークに接続された数千台のサーバーマシンで構成されています。サーバーボードとネットワーク機器の両方を Google がカスタム設計しています。提携するベンダーは入念に調査し、コンポーネントを慎重に選定したうえで、ベンダーと連携してそれらのコンポーネントが提供するセキュリティ特性を監査および検証しています。また、現在サーバーと周辺機器の両方にデプロイされているハードウェア セキュリティ チップを含むカスタムチップも設計しています。これらのチップにより、正規の Google 端末をハードウェア レベルで確実に特定して認証することができます。

ブートスタックとマシン ID のセキュリティ

Google サーバーマシンでは、正しいソフトウェア スタックを起動するためにさまざまな技術を利用しています。BIOS、ブートローダー、カーネル、基本オペレーティング システム イメージなどの下位コンポーネント

Google データセンターは、ローカル ネットワークに接続された数千台のサーバーマシンで構成されています。サーバーボードとネットワーク機器の両方を Google がカスタム設計しています。

に対しては暗号署名を使用しています。これらの署名はブートまたは更新ごとに検証することができます。コンポーネントはすべて Google が管理、構築、強化しています。Google では、新しい世代のハードウェアを使用して継続的なセキュリティ強化に努めています。たとえば、サーバー設計の世代に応じて、ブートチェーンの信頼の根拠を、ロック可能なファームウェア チップ、Google が制作したセキュリティ コードを実行するマイクロコントローラ、上記の Google が設計したセキュリティ チップのいずれかに置くようにしています。

データセンター内の各サーバーマシンには、信頼のハードウェア根拠とマシンが起動時に使用したソフトウェアに関連付けることが可能な固有の ID が割り当てられています。この ID は、マシン上の下位管理サービスとの間でやり取りされる API 呼び出しの認証に使用されます。

Google では、サーバーが最新バージョンのソフトウェア スタック（セキュリティ　パッチを含む）を実行することを保証し、ハードウェアとソフトウェアの問題を検出して診断し、必要に応じてサービスからマシンを除外する自動システムを構築しました。

サービスのデプロイのセキュリティ

ここでは、基本のハードウェアとソフトウェアから、インフラストラクチャへのサービスのデプロイのセキュリティを確保するところまでを説明します。ここで言う「サービス」とは、デベロッパーが制作し、Gmail　SMTP サーバー、BigTable ストレージ サーバー、YouTube 動画トランスコーダ、カスタム アプリケーションを実行する App Engine サンドボックスなどのインフラストラクチャ上で実行するアプリケーション　バイナリのことです。必要な規模のワークロードを処理するために何千台ものマシンが同じサービスのコピーを実行する場合もあります。インフラストラクチャ上で動作するサービスは、Borg　という名前のクラスタ　オーケストレーション サービスによって管理されます。

後述するように、インフラストラクチャはその上で動作しているサービス間の信頼を前提としません。つまり、インフラストラクチャは、基本的に、マルチテナントとして設計されています。

サービス ID、整合性、分離

サービス間通信用のアプリケーション　レイヤでは暗号認証および承認を使用しています。これにより、管理者とサービスが自然に認識できるような抽象化レベルと粒度で強力なアクセス制御が提供されます。

主要なセキュリティ　メカニズムとして、内部ネットワークのセグメント化またはファイアウォール化に依存していない代わりに、追加のセキュリティ　レイヤとして、IP　スプーフィングを回避するための入口フィルタリングと出口フィルタリングをネットワーク内のさまざまなポイントで使用しています。このアプローチは、ネットワークのパフォーマンスと可用性の最大化にも役立ちます。

インフラストラクチャ上で動作する各サービスには、サービス　アカウント ID が関連付けられます。サービスには、他のサービスとリモートプロシージャ コール（RPC）を送受信するときにその ID を証明するための暗号認証情報が付与されます。この　ID　は、クライアントが意図した正しいサーバーと通信していることを保証するためと、サーバーがメソッドとデータへのアクセスを特定のクライアントに制限するために使用されます。

Google のソースコードは中央レポジトリに保存されています。そこでは、最新バージョンのサービスと古いバージョンのサービスの両方を監査できます。加えて、インフラストラクチャは、サービスのバイナリをレビュー、チェックイン、テストが完了している特定のソースコードからビルドするように設定できます。このようなコードレビューには制作者以外に少なくとも 1 人のエンジニアの検査と承認が必要であり、さらに、どのシステムにおいてもコードを変更するにはそのシステムの所有者の承認が義務づけられています。これらの要件により、インサイダーや敵対者がソースコードに悪意のある変更を加えないよう制限され、サービスからそのソースまでの監査証跡も提供されます。

サービスを同じマシン上で動作している他のサービスから保護するためのさまざまな分離テクニックとサンドボックス化テクニックを使用しています。これらのテクニックには、正規の　Linux　ユーザーの分離、言語とカーネルベースのサンドボックス、ハードウェア仮想化が含まれます。通常は、よりリスクの高いワークロードに対してより多くのレイヤの分離を使用します。たとえば、ユーザーが指定したデータに対して複雑なファイル形式コンバータを実行する場合や、Google　App　Engine　や　Google Compute Engine などのプロダクトに対してユーザーが指定したコードを

サービス間通信用のアプリケーション　レイヤでは暗号認証および承認を使用しています。これにより、管理者とサービスが自然に認識できるような抽象化レベルと粒度で、強力なアクセス制御が提供されます。

実行する場合です。追加のセキュリティ境界として、クラスタ オーケストレーション サービスや一部の鍵管理サービスなどの非常に機密性の高いサービスを専用のマシン上で排他的に実行できます。

サービス間アクセスの管理

サービスの所有者は、インフラストラクチャが提供するアクセス管理機能を使用して、通信可能な他のサービスを正確に指定することができます。たとえば、あるサービスが他のサービスの特定のホワイトリストに一部の API のみを提供する場合です。このサービスは、許可されたサービス アカウント ID のホワイトリストを使用して設定することができ、そのアクセス制限はその後にインフラストラクチャによって自動的に適用されます。

サービスの所有者は、インフラストラクチャが提供するアクセス管理機能を使用して、通信可能な他のサービスを正確に指定することができます。

サービスにアクセスする Google のエンジニアにも個別の ID が発行されるため、サービスにはそのアクセスの許可と拒否も同様に設定できます。この種の ID のすべて（マシン、サービス、社員）が、インフラストラクチャが維持するグローバルな名前空間内に存在します。後述するように、エンドユーザー ID は別に処理されます。

インフラストラクチャは、承認チェーン、ロギング、通知を含む、これらの内部 ID のための豊富な ID 管理ワークフロー システムを提供します。たとえば、これらの ID は、あるエンジニアが他のエンジニア（グループの管理者でもある）の承認が必要なグループに対する変更を申し込むことができる二者管理を可能にするシステムを介してアクセス制御グループに割り当てることができます。このシステムを使用すれば、セキュアなアクセス管理プロセスを、インフラストラクチャ上で動作する何千ものサービスに拡張することができます。

自動 API レベル アクセス制御メカニズムに加えて、インフラストラクチャは、中央の ACL とグループのデータベースから読み取る機能もサービスに与えるため、必要に応じて、カスタムのきめ細かいアクセス制御を実装することができます。

サービス間通信の暗号化

前のセクションで説明した RPC 認証および承認機能に加えて、インフラストラクチャは、ネットワーク上の RPC データの暗号プライバシーと整合性も提供します。これらのセキュリティ機能を HTTP などの他の

アプリケーション レイヤ プロトコルでも利用できるように、これらをインフラストラクチャの RPC メカニズム内にカプセル化しています。つまり、これによりアプリケーション レイヤが分離され、ネットワーク パスのセキュリティに依存する必要がなくなります。ネットワークが不安定になったり、ネットワーク端末が侵害されたりしても、暗号化されたサービス間通信をセキュアなまま維持することができます。

サービスは、インフラストラクチャ RPC ごとに必要な暗号保護のレベルを設定することができます（たとえば、データセンター内部の低値データに対しては整合性レベルの保護を設定するだけです）。非公開の WAN リンクへの不正アクセスを試みる高度な知識を持った攻撃者から保護するために、インフラストラクチャはデータセンター間を WAN 経由で移動するすべてのインフラストラクチャ RPC トラフィックを自動的に暗号化します。サービスから明示的に設定する必要はありません。Google では、このデフォルトの暗号化をデータセンター内のすべてのインフラストラクチャ RPC トラフィックに拡張可能にするハードウェア暗号アクセラレータのデプロイを開始しました。

非公開の WAN リンクへの不正アクセスを試みる高度な知識を持った攻撃者から保護するために、データセンター間を WAN 経由で移動するすべてのインフラストラクチャ RPC トラフィックが自動的に暗号化されるようになっています。

エンドユーザー データのアクセス管理

標準的な Google サービスは、エンドユーザーのために何かをするように作られています。たとえば、エンドユーザーは、Gmail 上に自分のメールを保存しておくことができます。Gmail などのアプリケーションとエンドユーザーの相互作用は、インフラストラクチャ内の他のサービスにも及びます。そのため、たとえば、Gmail サービスは、エンドユーザーのアドレス帳にアクセスするために連絡先サービスから提供される API を呼び出すことができます。

Gmail サービス（または連絡先サービスが許可する他の特定のサービス）からの RPC リクエストのみが許可されるように連絡先サービスを設定できることは前のセクションで確認しました。

ただし、これは、広範囲に及ぶ権限の 1 つにすぎません。この権限の範囲内で、Gmail サービスはいつでも任意のユーザーの連絡先を要求することができます。

Gmail サービスは、特定のエンドユーザーの代わりに連絡先サービスに RPC リクエストを発行するため、インフラストラクチャは Gmail サービスに RPC の一部として「エンドユーザー権限チケット」を提示する機能を提供します。このチケットは、Gmail サービスが特定のエンドユーザーの

代わりにリクエストを処理していることを証明するものです。これにより、連絡先サービスは、チケット内で指定されたエンドユーザーに関するデータのみを返す安全保護対策を実装することができます。

インフラストラクチャは、これらの「エンドユーザー権限チケット」を発行する中央のユーザー アイデンティティ サービスを提供します。エンドユーザーのログインは、中央のアイデンティティ サービスで検証されます。その後で、このサービスが Cookie や OAuth トークンなどのユーザー認証情報をユーザーのクライアント端末に発行します。それ以降のクライアント端末から Google へのすべてのリクエストは、そのユーザー認証情報を提示する必要があります。

サービスがエンドユーザー認証情報を受け取ると、その認証情報を検証のために中央のアイデンティティ サービスに渡します。エンドユーザー認証情報が正しく検証されると、中央のアイデンティティ サービスがリクエストに関連した RPC に使用可能な有効期限の短い「エンドユーザー権限チケット」を返します。この例では、「エンドユーザー権限チケット」を取得するサービスが Gmail サービスであり、そこからチケットが連絡先サービスに渡されます。それ以降は、すべてのカスケード呼び出しに対して、呼び出し先への呼び出しサービスが RPC 呼び出しの一部として「エンドユーザー権限チケット」を継承できます。

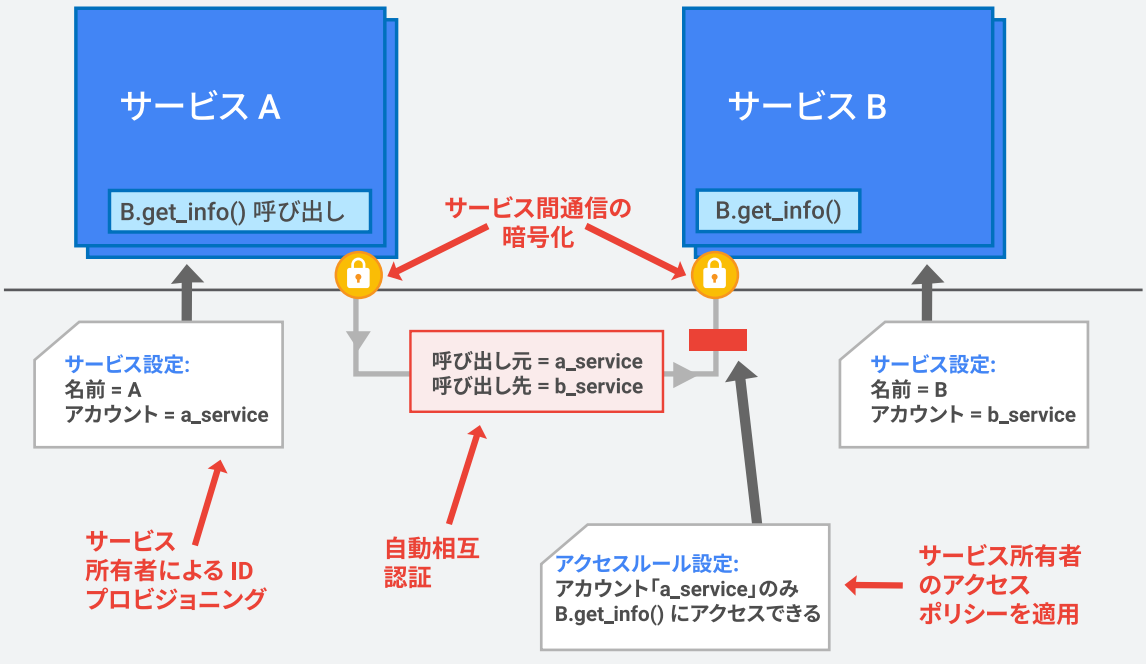


図 2.
サービス ID とアクセス管理：

インフラストラクチャは、サービスID、自動相互認証、暗号化されたサービス間通信、サービス所有者によって定義されたアクセス ポリシーの適用を可能にします。

データ ストレージのセキュリティ

ここまで、サービスのデプロイにおけるセキュリティについて説明してきました。ここからは、インフラストラクチャのセキュアなデータストレージの実装についての説明に移ります。

保存時の暗号化

Google のインフラストラクチャは、BigTable や Spanner などのさまざまなストレージ サービスと中央の鍵管理サービスを提供します。Google 上のほとんどのアプリケーションは、これらのストレージ サービスを介して間接的に物理ストレージにアクセスします。ストレージ サービスは、中央の鍵管理サービスから取得した鍵を使用して、物理ストレージに書き込まれる前のデータを暗号化するように設定できます。この鍵管理サービスは、自動鍵ローテーションをサポートし、豊富な監査ログを提供し、前述のエンドユーザー権限チケットと統合して、鍵を特定のエンドユーザーにリンクします。

アプリケーション　レイヤで暗号化を実行すると、インフラストラクチャは、ストレージの下位レベルでの潜在的な脅威（悪意のあるディスクファームウェアなど）からインフラストラクチャ自体を分離することができます。つまり、インフラストラクチャは、追加の保護レイヤも実装しています。Google　では、ハードドライブと　SSD　内のハードウェア暗号化サポートを有効にし、すべてのドライブをそのライフサイクルを通して細かく追跡しています。廃棄予定の暗号化されたストレージ　デバイスは物理的に管理下から外される前に、2 回の独立した検証を含む多段階プロセスを使用してクリーニングされます。このワイプ手順を通過していないデバイスは、オンプレミスで物理的に破壊（細断など）されます。

データの削除

Google におけるデータの削除は、ほとんどの場合、データを完全に削除するのではなく、特定のデータを「削除予定」としてマークすることから始まります。これにより、お客様が実施したのか、内部的なバグや処理エラーが原因なのかに関係なく、意図しない削除からの回復が可能になります。「削除予定」としてマークされたデータは、サービス固有のポリシーに従って削除されます。

非公開の WAN リンクへの不正アクセスを試みる高度な知識を持った攻撃者から保護するために、データセンター間を WAN 経由で移動するすべてのインフラストラクチャ RPC トラフィックが自動的に暗号化されるようになっています。

エンドユーザーが自分のアカウント全体を削除した場合は、インフラストラクチャがアカウントが削除されたことをエンドユーザー データを処理するサービスに通知します。その後で、サービスは、削除されたエンドユーザー アカウントに関連付けられたデータを削除するようにスケジュールすることができます。この機能を使用すれば、サービスのデベロッパーは、エンドユーザー制御を簡単に実装することができます。

インターネット通信のセキュリティ

ここまで、インフラストラクチャ上でサービスを保護する方法について説明してきました。このセクションでは、インターネットとこれらのサービス間の通信を保護する方法に関する説明に移ります。

前述したように、インフラストラクチャは、LAN や WAN を介して相互接続された物理マシンの大規模なセットで構成されており、サービス間通信のセキュリティは、ネットワークのセキュリティに依存していません。ただし、インフラストラクチャをインターネットから非公開の IP 空間に分離することにより、マシンのサブセットを直接外部のインターネット トラフィックに公開するだけで、サービス拒否（DoS）攻撃に対する防御などの追加の保護をより簡単に実装できます。

Google Front End サービス

サービスをインターネット上で利用可能にするには、それをGoogle Front End（GFE）と呼ばれるインフラストラクチャ サービスに登録する必要があります。GFE は、すべての TLS 接続が、正しい証明書を使用し、完全な前方秘匿性のサポートなどのベスト プラクティスに従って終端されることを保証します。加えて、GFE は、サービス拒否攻撃に対する防御（詳細は後述）を適用します。その後で、GFE が前述の RPC セキュリティプロトコルを使用してサービスにリクエストを転送します。

実際には、外部に公開する内部サービスでは、GFE がスマートなリバースプロキシ フロントエンドとして使用されます。このフロントエンドは、パブリック DNS 名のパブリック IP ホスティング、サービス拒否（DoS）攻撃に対する防御、TLS終端を提供します。GFE は、他のサービスと同様のインフラストラクチャ上で動作するため、着信リクエストの量に合わせてスケーリングできます。

Google Front End では、すべての TLS 接続が正しい証明書を使用し、完全な前方秘匿性のサポートなどのベストプラクティスに従って終端されることが保証されています。

サービス拒否（DoS）攻撃に対する防御

インフラストラクチャの規模が大きいために、Google では多くの DoS 攻撃を単純に吸収することができます。つまり、GFE の背後で動作しているサービスに対する DoS の影響のリスクを大幅に低減するマルチティアでマルチレイヤの DoS 防御が施されています。

Google のバックボーンは、データセンターのいずれかに外部接続を配信した後で、ハードウェアとソフトウェアの負荷分散の複数のレイヤを通過します。これらのロードバランサは、インフラストラクチャ上で動作している中央の DoS サービスへの受信トラフィックに関する情報を報告します。中央の DoS サービスは、DoS 攻撃が行われていることを検出すると、攻撃に関連付けられたトラフィックを破棄または抑制するようにロードバランサを設定することができます。

次のレイヤでは、GFE インスタンスが受信中のリクエストに関する情報も中央の DoS サービスに報告します。この情報には、ロードバランサが把握していないアプリケーションレイヤの情報が含まれます。その後で、中央の DoS サービスが、攻撃トラフィックを破棄または抑制するように GFE インスタンスを設定することもできます。

ユーザー認証

DoS に対する防御の次の防御レイヤは、中央のアイデンティティ サービスからもたらされます。このサービスは、通常、Google のログインページとしてエンドユーザーに提示されます。単純なユーザー名とパスワードを要求するのではなく、サービスは、ユーザーに対して、過去に同じ端末または同様の場所からログインしたことがあるかどうかなどのリスク要因に基づいて自動的に追加情報を要求します。ユーザーの認証後は、アイデンティティ サービスが、以降の呼び出しに使用可能な Cookie や OAuth トークンなどの認証情報を発行します。

ユーザーは、ログイン時に、OTP やフィッシング耐性のあるセキュリティ キーなどの 2 つ目の要素を採用することもできます。Google にとっても大きなメリットがあることを確認するために、FIDO Alliance で複数の端末ベンダーと協力して Universal 2nd Factor（U2F）オープン スタンダードを策定しました。現在、これらの端末は市場で入手可能であり、他の主要なウェブサービスも U2F のサポートを導入し始めています。

インフラストラクチャの規模が大きいために、Google では多くの DoS 攻撃を単純に吸収することができます。つまり、GFE の背後で動作しているサービスに対する DoS 攻撃のリスクを大幅に低減する、マルチティアでマルチレイヤの DoS 防御が施されています。

オペレーション セキュリティ

ここまで、インフラストラクチャに組み込まれているセキュリティ設計についてと、RPC 上でのアクセス制御といったセキュア オペレーションの仕組みの一部について説明してきました。

ここからは、インフラストラクチャの実際の運用におけるセキュリティの説明に移ります。Google では、セキュリティの万全なインフラストラクチャ ソフトウェアを作成し、社員のマシンと認証情報を保護し、内部と外部両方の攻撃者からのインフラストラクチャに対する脅威を防ぎます。

安全なソフトウェアの開発

前述した中央のソース管理機能と二者レビュー機能に加えて、デベロッパーが特定のクラスのセキュリティ バグを発生させないようにするためのライブラリも提供しています。たとえば、ウェブアプリの XSS 脆弱性を排除するライブラリとフレームワークが用意されています。また、ファザーなどのセキュリティバグを自動的に検出するための自動ツール、静解析ツール、ウェブ セキュリティ スキャナも用意されています。

最終チェックとして、リスクが低い機能の迅速な選別から、最もリスクが高い機能の詳細設計および実装レビューまでにおよぶ手動セキュリティ レビューを使用しています。これらのレビューは、ウェブ セキュリティ、暗号化、オペレーティング システム セキュリティの各専門家を交えたチームによって実施されます。また、レビューは、新しいセキュリティ ライブラリ機能につながることもあれば、他の将来のプロダクトに適用可能な新しいファザーにつながることもあります。

加えて、インフラストラクチャやアプリケーションのバグを発見して報告した人に報奨金を出す脆弱性報奨金プログラムを運営しています。これまで、このプログラムで数百万ドルの報奨金が支払われています。

また、Google では、使用しているすべてのオープン ソース ソフトウェアのゼロデイ エクスプロイトなどのセキュリティ上の問題の発見と、それらの問題のアップストリームに全力で取り組んでいます。たとえば、OpenSSL Heartbleed バグは Google で発見されましたし、Linux KVM ハイパーバイザの CVE とセキュリティ バグの修正の最大の提出者でもあります。

Google では、インフラストラクチャやアプリケーションのバグを発見して報告した人に報奨金を出す脆弱性報奨プログラムを実施しています。

社員の端末と認証情報の安全の確保

Google では、社員の端末と認証情報を侵害から守る保護活動と、潜在的な情報漏洩や違法なインサイダー行為を発見するための監視活動に多くの投資を行っています。これは、インフラストラクチャが安全に運用されていることを保証するための投資の重要な部分です。

長年にわたって、巧妙なフィッシングが社員を標的とした手段でした。この脅威から保護するために、フィッシングされる可能性のある OTP 第 2 要素を、社員アカウントに対する U2F 互換セキュリティ キーの必須使用に置き換えました。

社員がインフラストラクチャの運用に使用するクライアント端末の監視に多大な投資を行っています。これらのクライアント端末のオペレーティング システム イメージがセキュリティ パッチを含む最新版であることを保証し、インストール可能なアプリケーションを管理しています。加えて、ユーザーがインストールしたアプリ、ダウンロード、ブラウザの拡張機能、ウェブから閲覧されたコンテンツの法人顧客に対する適合性をスキャンするためのシステムを導入しています。

アクセス権限を付与する主な基準は、社内の LAN 上に存在するかどうかではありません。代わりに、想定されるネットワークや地理的な場所で正しく管理された端末からアクセスしている特定のユーザーにのみ内部アプリケーションを公開できるようにするアプリケーション レベルのアクセス管理コントロールを使用しています（詳細については、BeyondCorp に関する追加情報をご覧ください）。

インサイダー リスクの低減

Google では、インフラストラクチャへの管理アクセス権が付与された社員の活動を強制的に制限し、積極的に監視しているほか、同じタスクを安全で管理された方法で自動的に実行する処理を提供することにより、特定のタスクに対する特権アクセスの必要性を排除する努力を続けています。たとえば、特定のアクションの実施に二者の承認を必須とする処置や、機密情報を公開することなくデバッグすることが可能な制限付き API の導入などです。

Compute Engine 制御プレーン API に対するエンドユーザー認証は、ハイジャック検出などのセキュリティ機能を提供する Google の集中型アイデンティティ サービスを介して実施されます。承認は、中央の Cloud IAM サービスを使用して実施されます。

侵入検知

Google では、個々の端末上のホストベースの信号、インフラストラクチャ内のさまざまなモニタリング ポイントからのネットワーク ベースの信号、インフラストラクチャ サービスからの信号を統合する高度なデータ処理パイプラインを導入しています。これらのパイプライン上に構築されたルールとマシンインテリジェンスから、可能性のあるインシデントの警告が運用セキュリティ エンジニアにもたらされます。Google の調査およびインシデント対応チームは、これらの潜在的なインシデントを年中無休で選別、調査、対応しています。Google では、検出メカニズムと対応メカニズムの有効性を評価して改善するための Red Team 訓練を実施しています。

信号をモニタリングするパイプライン上に構築されたルールとマシン インテリジェンスから、可能性のあるインシデントの警告が運用セキュリティ エンジニアに通知されます。

Google Cloud Platform (GCP) の保護

このセクションでは、Google のパブリック クラウド インフラストラクチャである GCP が、基礎となるインフラストラクチャのセキュリティからメリットを得ている様子を説明します。Google Compute Engine サービスを例として取り上げ、インフラストラクチャ上に構築されたサービス固有のセキュリティ強化について詳しく説明します。

GCEを使用すれば、お客様は Google のインフラストラクチャ上で独自の仮想マシンを実行することができます。GCE 実装は、いくつかの論理コンポーネントで構成されます。注目すべき主なコンポーネントは管理制御プレーンと仮想マシン自体です。

管理制御プレーンは、外部 API サーフェスを公開し、仮想マシンの作成や移行などのタスクをオーケストレートします。このプレーンは、インフラストラクチャ上のさまざまなサービスとして動作するため、セキュアなブートチェーンなどの基本的な整合性機能を自動的に利用します。個々のサービスはそれぞれの内部サービス アカウントの下で実行されるため、すべてのサービスには、制御プレーンの残りの部分にリモート プロシージャ コール (RPC) を発行するときに必要になる権限のみを付与できます。前に述べたように、これらのすべてのサービスのコードが中央のGoogle ソースコード レポジトリに格納され、このコードと最終的にデプロイされるバイナリとの間の監査証跡が生成されます。管理制御プレーンは、外部 API サーフェスを公開し、仮想マシンの作成や移行などのタスクをオーケストレートします。このプレーンは、インフラストラクチャ上のさまざまなサービスとして動作するため、セキュアなブートチェーンなどの基本的な整合性機能を自動的に利用します。個々のサービスはそれぞれの内部サービス アカウントの下で実行されるため、すべてのサービスには、制御プレーンの残りの部分にリモート プロシージャ コール (RPC) を発行するときに必要になる権限のみを付与できます。前に述べたように、これらのすべてのサービスのコードが中央の Google ソースコード レポジトリに格納され、このコードと最終的にデプロイされるバイナリとの間の監査証跡が生成されます。

GCE 制御プレーンは、GFE を介してその API を公開するため、サービス拒否（DoS）攻撃の防御や一元管理された SSL/TLS のサポートなどのインフラストラクチャ セキュリティ機能を利用します。Google Cloud Load Balancer サービスは、GFE 上に構築された、さまざまな種類の DoS 攻撃を緩和できるオプションです。お客様はこのオプションの使用を選択することで、GCE VM 上で動作するアプリケーションに対して同様の保護を手に入れることができます。

Compute Engine 制御プレーン API に対するエンドユーザー認証は、ハイジャック検出などのセキュリティ機能を提供する Google の集中型アイデンティティ サービスを介して実施されます。承認は、中央の Cloud IAM サービスを使用して実施されます。

GFE からその背後にある最初のサービスに流れるものとその他の制御プレーン サービス間を流れるものの両方の制御プレーンのネットワーク トラフィックは、自動的に、インフラストラクチャによって認証され、データセンター間を移動するたびに暗号化されます。加えて、インフラストラクチャは、データセンター内の制御プレーン トラフィックの一部を暗号化するようにも設定されています。

各仮想マシン（VM）は、関連する仮想マシン マネージャ（VMM）サービス インスタンスで動作します。インフラストラクチャは、これらのサービスに 2 つの ID を付与します。1 つの ID は、独自の呼び出し用の VMM サービス インスタンスによって使用され、もう 1 つの ID は、VMM がお客様の VM の代わりに発行する呼び出しに使用されます。これにより、VMM から着信した呼び出しに対する信頼をさらに分割することができます。

GCE 永続ディスクは、中央のインフラストラクチャ鍵管理システムによって保護された鍵を使用して保存中に暗号化されます。これにより、これらの鍵へのアクセスの自動ローテーションと中央監査が可能になります。

現在、お客様には、トラフィックを VM 間で送信する、インターネットにプレーンテキストで送信する、トラフィック用に選択した暗号化を実装する、の選択肢があります。Google では、お客様の VM 間トラフィックの WAN トラバーサル ホップ用の自動暗号化の展開を開始しました。前に述べたように、インフラストラクチャ内のすべての制御プレーン WAN トラフィックはすでに暗号化されています。将来的には、データセンター内の VM 間 LAN トラフィックも暗号化するために、前述のハードウェアで加速するネットワーク暗号化を利用する予定です。

Google Compute Engine（GCE）制御プレーンでは、Google Front End（GFE）を介してその API を公開するため、サービス拒否（DoS）攻撃の防御や一元管理された SSL/TLS のサポートなどのインフラストラクチャ セキュリティ機能を利用します。

VM に提供される分離は、オープンソース KVM スタックを使用したハードウェア仮想化に基づきます。コントロールとハードウェア　エミュレーション　スタックの一部をカーネル外部の非特権プロセスに移動することにより、KVMの特定の実装をさらに強化しました。また、ファジング、静解析、手動コードレビューなどのテクニックを使用して、KVM のコアを広範囲にテストしました。前に述べたように、KVM　にアップストリームされた最近公開された脆弱性の大半が Google から提出されたものです。

最後に、オペレーション　セキュリティ制御は、データへのアクセスがポリシーに従っていることの確認の重要な部分です。Google Cloud Platform の一部として、Compute Engine のお客様のデータの使用法は、顧客データポリシーの GCP 使用法に従っています。つまり、Google が、お客様にサービスを提供するために必要な場合を除き、お客様のデータにアクセスしたり、使用したりすることはありません。

まとめ

サービスをインターネット規模で安全に構築、デプロイ、運用するためにGoogleインフラストラクチャがどのように設計されているかについて説明しました。この中には、Gmail などの消費者向けサービスと企業向けサービスの両方が含まれています。加えて、Google Cloud の各プロダクトもこの同じインフラストラクチャ上に構築されています。

Googleでは、インフラストラクチャの保護に多額の投資をしており、セキュリティとプライバシー専門の数百人ものエンジニアが　Google　全域に配置されています。業界の権威として認識されているエンジニアも多数います。

これまで見てきたように、インフラストラクチャのセキュリティは、まずは物理コンポーネントとデータセンター、そしてハードウェアの供給元、次にブートのセキュリティ、サービス間通信のセキュリティ、保存データのセキュリティ、インターネットからサービスへの保護アクセス、そして最後に、オペレーションのセキュリティのために導入されている技術と人材によるプロセスというように、進行型の階層構造で設計されています。

Google では、インフラストラクチャの保護に多額の投資をしており、Google の全域に数百人を数えるセキュリティおよびプライバシー専門のエンジニアを擁しています。中には、業界の権威として認知されている人材もいます。

追加情報

特定の分野の詳細については、以下の資料をご覧ください。

1. データセンターの物理的セキュリティ
   https://goo.gl/WYlKGG
2. クラスタの管理とオーケストレーションの設計
   http://research.google.com/pubs/pub43438.html
3. ストレージ暗号化機能と顧客対応 GCP 暗号化機能
   https://cloud.google.com/security/encryption-at-rest/
4. BigTable ストレージ サービス
   http://research.google.com/archive/bigtable.html
5. Spanner ストレージ サービス
   http://research.google.com/archive/spanner.html
6. ネットワーク負荷分散のアーキテクチャ
   http://research.google.com/pubs/pub44824.html
7. 企業セキュリティに対する BeyondCorp のアプローチ
   http://research.google.com/pubs/pub43231.html
8. セキュリティ キーと Universal 2nd Factor (U2F) 標準を使用したフィッシング対策
   http://research.google.com/pubs/pub45409.html
9. Google 脆弱性報奨金プログラムの詳細
   https://bughunter.withgoogle.com/
10. GCP 上での HTTP とその他の負荷分散オファリングの詳細
    https://cloud.google.com/compute/docs/load-balancing/
11. GCP 上での DoS 防御のベスト プラクティスの詳細
    https://cloud.google.com/files/GCPDDoSprotection-04122016.pdf
12. Google Cloud Platform の顧客データポリシーの使用
    https://cloud.google.com/terms/
13. G Suite (Gmail、ドライブなど) でのアプリケーション セキュリティとコンプライアンスの詳細)
    https://goo.gl/3J20R2

Google Cloud での保存時の暗号化

Google Cloud Platform 暗号化に関するホワイトペーパー

Google Cloud

Google Cloud

目次

これは、Google が実践している暗号化によるユーザーデータ保護について論じた 2 つのホワイトペーパーの 2 つ目にあたります。もう 1 つは G Suite 暗号化に関するホワイトペーパーです。両方ともお読みになると、Google での暗号化の使用に関する知識を増やすうえで役立ちます。

このホワイトペーパーでは、Google の暗号鍵の階層と信頼のルートについて詳しく説明するとともに、特定の GCP サービスにおける保存データの暗号化の粒度に関する情報を提供します（転送中のデータの暗号化は本書の対象外です）。

すべての Google プロダクトについて、Google は顧客データの厳重な保護に努め、採用しているセキュリティ保護の方式についても可能な限り透明性を確保するよう努めています。

このドキュメントの内容は 2017 年 4 月時点のもので、作成時点の状況を表しています。お客様の保護の継続的な改善のために、Google Cloud Platform のセキュリティポリシーとシステムは変更される場合があります。

CIO レベルの概要

- Google では、複数の暗号化レイヤを使用して、Google Cloud Platform プロダクト内のお客様の保存データを保護しています。
- Google Cloud Platform では、1 つ以上の暗号化メカニズムを使用して、保存されているお客様のコンテンツを暗号化しています。お客様による操作は必要ありません。いくつかの例外があります。詳細については、このドキュメントを参照してください。
- 保存するデータは複数のチャンクに分割されます。各チャンクは一意のデータ暗号鍵で暗号化されます。これらのデータ暗号鍵はデータとあわせて保存され、Google の Key Management Service で独占的に保存され、使用される暗号鍵で暗号化（「ラップ」）されます。Google の Key Management Service は冗長的で、グローバルに分散しています。
- Google Cloud Platform に保存されるデータは AES256 または AES128 を使用してストレージ レベルで暗号化されます。
- Google は共通暗号ライブラリ「Keyczar」を使用して、ほぼすべての Google Cloud Platform プロダクトで継続的に暗号化を実施しています（Keyczar のオープンソース バージョンにはセキュリティの問題があることがわかっているため、Google では使用されません）。この共通ライブラリは広範囲からアクセスできるため、この厳密に管理および審査されたコードを実装、維持するには、少人数の暗号作成者チームが 1 チームあれば十分です。

はじめに

多くの人や企業にとって、セキュリティはパブリッククラウド　ベンダーを選ぶための重要な基準です。Googleでは、セキュリティを最も重要な項目と見なしています。Google はセキュリティとプライバシーを重要視し、データ保護のためのたゆまぬ努力を続けています。これはインターネット上で流れるデータについても、Google のデータセンター間でデータを移動する場合でも、Google のサーバーに保存されたデータについても同様です。

Google　の総合的なセキュリティ戦略の中核をなすのは、データ通信時および保存時の暗号化です。これにより、データにアクセスできるのは暗号鍵への監査済みアクセス権が承認されている役割とサービスだけになります。このドキュメントでは、Google Cloud Platform での保存時の暗号化の手法と、Google がこの暗号化によって情報をどのようにして安全に保護しているかを説明します。

このドキュメントは、現在 Google Cloud Platform を使用している、または使用を検討している CISO やセキュリティ運用チームを対象としています。このドキュメントは、「はじめに」の項を除き、暗号化と初歩的な暗号についての基本事項を理解していることを前提としています。

暗号化とは

暗号化とは、読み取り可能なデータを入力（プレーンテキスト）として使用して、元のプレーンテキストの情報がほとんどないか、まったくない出力（暗号テキスト）に変換する処理のことです。使用する暗号化アルゴリズムは　AES　のように公開されているものですが、実行には秘密の鍵を使用します。暗号テキストを元の形に復号化するには、この鍵を使用する必要があります。Google では、暗号化を使用してデータの機密保持を行うことで整合性を維持しています。暗号テキストにアクセスできても、鍵がわからなければ内容を理解することも変更することもできません。暗号については、『Introduction to Modern Cryptography』も参照してください。

暗号化とは、読み取り可能なデータを入力 (プレーンテキスト) として使用して、元のプレーンテキストの情報がほとんどまたはまったくわからない出力 (暗号テキスト) に変換する処理のことです。

このホワイトペーパーでは、保存時の暗号化について説明しています。「保存時の暗号化」とは、ディスク（ソリッド ステート ドライブを含みます）またはバックアップ　メディアに保存されているデータを保護するための暗号化を指します。

暗号化による顧客データの安全性の向上

暗号化は広範囲なセキュリティ戦略の一部です。暗号化により、データ保護のためのより高度な防御を実現できます。暗号化を使用すると、万が一攻撃者がデータを入手したとしても、暗号鍵がなければデータにアクセスすることはできません。データが保存されているストレージ　デバイスを攻撃者が入手しても、そのデータを理解することも復号化することもできません。

保存時の暗号化により、ハードウェアとソフトウェア スタックの下位レイヤが実質的に「切除」されるため、攻撃可能な箇所が削減されます。これらの下位レイヤが（デバイスへの物理的なアクセスなどにより）攻撃を受けても、適切な保護処理が行われていれば、これらのデバイスのデータには影響はありません。暗号化は「チョークポイント」にもなります。暗号鍵は一元管理されているため、データへのアクセスは必ずここを経由しなければならず、アクセスの監査も可能となります。

暗号化は、Google　が顧客データのプライバシーを確保するための重要なメカニズムです。これにより、コンテンツにアクセスしなくても、システムがバックアップなどの目的でデータを操作でき、エンジニアはインフラストラクチャのサポートができるようになります。

顧客データとして扱われるデータ

Google Cloud Platform 利用規約で定められているとおり、「顧客データ」とは Google Cloud Platform の顧客（またはその指示を受けた人物）により、その顧客のアカウントで使用する Google Cloud Platform サービスを介して直接的または間接的に Google に提供されたコンテンツを指します。顧客データには、顧客コンテンツと顧客メタデータがあります。

「顧客コンテンツ」は、Google Cloud Platform の顧客が自分で作成したデータまたは Google に提供したデータです。Google Cloud Storage に保存したデータ、Google Compute Engine で使用されるディスク スナップショット、Cloud IAM ポリシーなどが該当します。このドキュメントで主に取り上げるのは、顧客コンテンツの保存時の暗号化です。

「顧客メタデータ」とは、顧客データの残りの部分、つまり顧客コンテンツとして分類できないすべてのデータを指します。自動生成されたプロジェクト番号、タイムスタンプ、IP アドレスや、Google Cloud Storage のオブジェクトのバイトサイズ、Google Compute Engine のマシンタイプなどが該当します。メタデータは、進行中のパフォーマンスやオペレーションにとって妥当な範囲で保護されます。

顧客データとは、Google Cloud Platform の顧客またはその指示を受けた人物により、その顧客のアカウントで使用する Google サービスを介して直接的または間接的に Google に提供されるコンテンツを指します。

Google のデフォルトの暗号化

データ保存時の暗号化

暗号化レイヤ

Google では複数の暗号化レイヤを使用してデータを保護しています。複数の暗号化レイヤを使用することにより、冗長データ保護機能が追加され、アプリケーションの要件に基づいて最適な手法を選択できるようになります。

暗号鍵の階層と信頼のルート

Google の KMS は「KMS マスター鍵」と呼ばれるルート鍵で保護されています。この鍵は KMS のすべての KEK をラップします。この KMS マスター鍵は AES2565 で、これ自体が「ルート KMS」と呼ばれる別の鍵管理サービスに保存されています。ルート KMS にははるかに少ない数（12 個

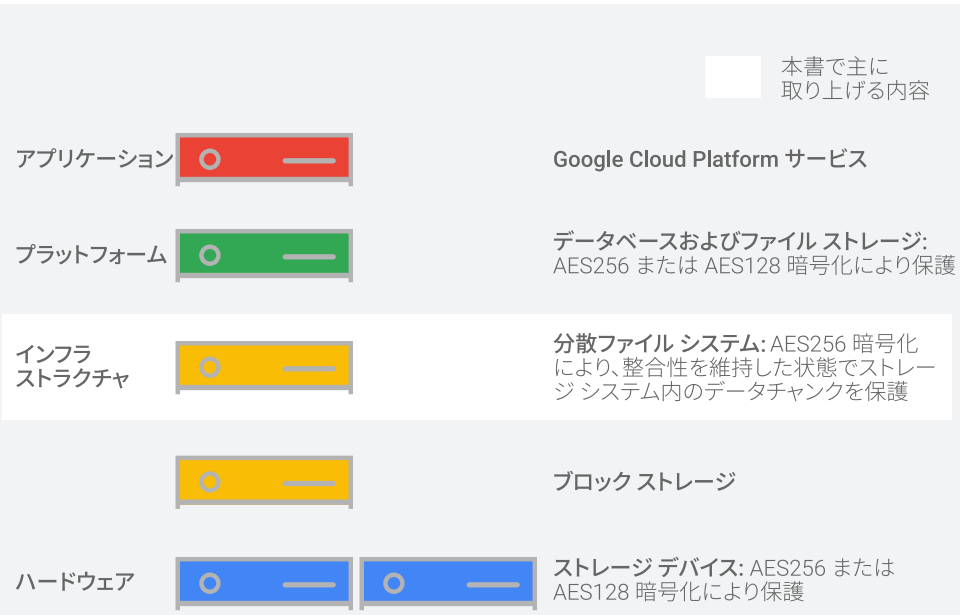


図 1:

Google Cloud Platform では保存されているデータを保護するために、複数の暗号化レイヤが使用されています。ほぼすべてのファイルで、分散ファイル システム暗号化またはデータベースおよびファイル ストレージ暗号化が使用されています。また、ほぼすべてのファイルでストレージ デバイス暗号化が使用されています。

程度）の鍵が保存されています。セキュリティ向上のため、ルートKMSは一般的な本番機では実行されず、各 Google データセンターの専用機でのみ実行されます。

Google では、データをディスクに書き込む前にデータを暗号化しています。暗号化は Google のすべてのストレージ システムに最初からある機能であり、後から追加されるものではありません。

各データチャンクには一意の識別子があります。アクセス制御リスト（ACL）により、各チャンクは承認された役割によって運用されている Google サービスのみで復号化できます。これらのサービスにはその時点でアクセスが許可されます。これにより、承認なくデータにアクセスすることはできなくなり、データのセキュリティとプライバシーが強化されます。

各チャンクは Google のストレージ システム間に分散され、バックアップと障害復旧のために暗号化して複製されます。悪意のあるユーザーが顧客データにアクセスしようとする場合、(1) 目的のデータに該当するすべてのストレージ チャンクと、(2) そのチャンクに対応する暗号鍵がわかっていて、アクセス権を持っている必要があります。

各データチャンクは個別の暗号鍵を使用してストレージ レベルで暗号化されます。2 つのチャンクの暗号鍵が同じになることはありません。これはその 2 つのチャンクが同じ Google Cloud Storage オブジェクトの一部であっても、同じ顧客が所有するものであっても、また同じマシンに保存されていても同様です。

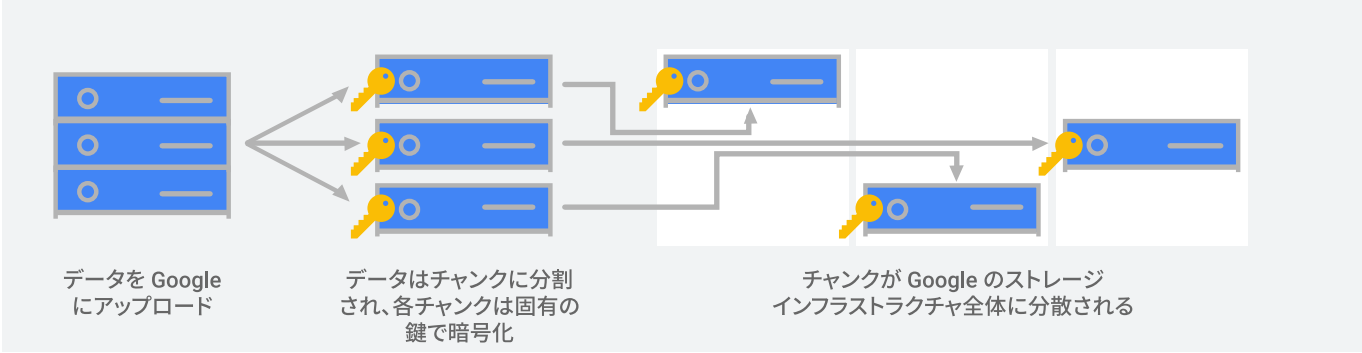


図 2:

Google のデータは保存のためにチャンクに分割され暗号化されます。

GoogleはAESアルゴリズムを使用して保存時の暗号化を行っています。AES が広く利用されているのは、(1) AES256 と AES128 の両方が長期保存用としてアメリカ国立標準技術研究所（NIST）により推奨されていて（2015 年 11 月現在）、(2) AES が顧客のコンプライアンス要件の一部となっていることが多いためです。

[1] Cloud Datastore、App Engine、Cloud Pub/Sub のデータチャンクには 2 件の顧客のデータが存在することがあります。サービスによるデータ暗号化の粒度に関するセクションを参照してください。

Google Cloud Storage で保存されているデータは AES を使用してストレージレベルで暗号化されます。ほとんどの場合、Galois/Counter Mode (GCM) が使用されます。これは Google が管理する BoringSSL ライブラリ で実装されています。このライブラリは、 OpenSSL で数多くの欠陥が発見された後、内部使用のために OpenSSL を基に作成されました。選択の場合、AES は認証用に HMAC (ハッシュ化メッセージ認証コード) による CBC (暗号ブロック チェーン) モードで使用されます。複製された一部のファイルでは、AES は HMAC による CTR (カウンタ) モードで使用されます (アルゴリズムの詳細についてはこのドキュメントで後述します) 。その他の Google Cloud Platform プロダクトでは、AES はさまざまなモードで使用されます。

ストレージ デバイス レイヤでの暗号化

上記のストレージ システム レベルの暗号化に加えて、多くの場合、データはストレージ デバイスレベルでも暗号化されます。この暗号化には少なくとも、ハードディスク (HDD) では AES128、新しいソリッド ステート ドライブ (SSD) では AES256 が使用され、また個別のデバイスレベルの鍵が使用されます (これはストレージ レベルでのデータの暗号化に使用した鍵とは異なります) 。デバイスの交換が進めば、デバイスレベルの暗号化では AES256 のみが使用されるようになります。

バックアップの暗号化

Google のバックアップ システムでは、バックアップ プロセス全体でデータが暗号化されています。この方法により、必要でないときにプレーンテキスト データが読み取られることがなくなります。

さらに、このバックアップ システムでは独自のデータ暗号鍵 (DEK) を使用して各バックアップ ファイルが個別に暗号化されます。この DEK は、Google の KMS (鍵管理サービス) に保存されている鍵と、バックアップ時にランダムに生成されるファイル別のシードを使用して生成されます。また、バックアップ時にはすべてのメタデータに対して別の DEK が使用されます。この DEK も Google の KMS に保存されています (鍵管理の詳細についてはこの後のセクションで説明します) 。

データが保存時に暗号化されない場合

Google Cloud Platform は顧客による操作がなくても、1 つ以上の暗号化メカニズムを使用して、保存されている顧客コンテンツを暗号化します。ただし、次のような例外があります。

- Google Compute Engine の仮想マシンのシリアル コンソール ログ。これについては現在修正中です。

顧客が Google Cloud Platform で保存したコンテンツは、顧客が操作をしなくても 1 つ以上の暗号化メカニズムを使用して暗号化されます (ただし、ごく一部の例外があります) 。

- プロセスで予期しない障害が発生した場合にローカル ドライブに書き込まれるコアダンプ。これについては現在修正中です。
- ローカル ディスクに書き込まれるデバッグログ。これについては現在修正中です。
- ストレージ　システムが使用する一時ファイル。これについては現在修正中です。
- 顧客コンテンツや顧客メタデータが含まれる可能性がある一部のログ。今後修正予定です。

このデータは Google の他のすべてのセキュリティ インフラストラクチャにより広範囲で保護されています。ほぼすべての場合で、ストレージレベルの暗号化により保護されます。

鍵管理

データ暗号鍵、鍵暗号鍵、Google の鍵管理サービス

チャンクのデータを暗号化するための鍵は「データ暗号鍵（DEK）」と呼ばれています。Google で使用する鍵は膨大な数にのぼり、また低レイテンシと高可用性を実現する必要があるため、これらの鍵は暗号化対象のデータの近くに保存されます。DEK は「鍵暗号鍵（KEK）」を使用して暗号化（「ラップ」）されます。Google Cloud Platform サービスごとに 1 つ以上の KEK があります。これらの KEK は Google の「鍵管理サービス（KMS）」に一括保存されています。これは鍵を保存するための専用のレポジトリです。KEK の数を DEK より少なくして、1 つの鍵管理サービスにより一括管理することにより、Google でのデータの保存と暗号化がスケール管理可能となり、データの追跡と制御を 1 か所からできるようになります。

Google Cloud Platform の顧客ごとに、共有されていないリソース[2]はデータチャンクに分割され、他の顧客用に使用された鍵とは別の鍵で暗号化されます[3]。これらの DEK は、同じ顧客が所有する同じデータの他の部分を保護する DEK とも異なります。

チャンク内のデータを暗号化するための鍵は「データ暗号鍵（DEK）」と呼ばれます。DEK は「鍵暗号鍵（KEK）」を使用して暗号化されます。KEK は Google の「鍵管理サービ（KMS）」（KEK 管理専用のレポジトリ）に一括保存されます。

[2] 共有リソース（この分離が実行されていない）の例として、Google Compute Engine の共有ベースイメージがあげられます。ここでは、複数の顧客が、1 つの DEK で暗号化された 1 つのコピーを参照しています。
[3] Cloud Datastore、App Engine、Cloud Pub/Sub に保存されたデータは例外で、同じ DEK で複数の顧客のデータを暗号化できます。サービスによるデータ暗号化の粒度に関するセクションを参照してください。

DEK はストレージ システムにより Google の共通暗号ライブラリを使用して生成されます。その後 KMS に送信され、そのストレージ システムの KEK でラップされます。ラップされた DEK はストレージ システムに戻され、データチャンクと一緒に保存されます。ストレージ システムが暗号化されたデータを取得する必要がある場合、ラップされた DEK を取得してから KMS に渡します。KMS はこのサービスが KEK の使用を承認されているか確認して、承認されていればラップを解除し、プレーンテキストの DEK をサービスに返します。サービスはこの DEK を使用してデータチャンクをプレーン テキストに復号化し、整合性を確認します。

データチャンクの暗号化に使用する KEK の大部分は KMS 内で生成され、残りはストレージ サービス内で生成されます。一貫性を確保するため、KEK はすべて Google の共通暗号ライブラリと Google が作成した乱数生成ツール（RNG）を使用して生成されます。この RNG は NIST 800-90A に基づいていて、AES256 KEK を生成します[4]。この RNG は Linux カーネルの RNG が基になっています。また、Linux カーネルの RNG は複数の独立したエントロピー ソースが基になっています。これにはデータセンター環境からのエントロピー イベントが含まれます。たとえば、ディスクシークの詳細な測定、パケット間の到着時間、また使用可能な場合は Intel の RDRAND 命令（Ivy Bridge 以降の CPU）などがこれに該当します。

Google Cloud Platform に保存されているデータは前述の通り AES256 または AES128 を使用する DEK により暗号化されます。また、Google Compute Engine の永続ディスクで新しく暗号化されるデータは、AES256 を使用して暗号化されます。DEK は、Google Cloud Platform サービスに応じて、AES256 または AES128 を使用する KEK によりラップされます。現在、Cloud サービスのすべての KEK を AES256 にアップグレードする作業を進めています。

Google の KMS は KEK 管理専用のサービスで、その他の目的では使用されません。このサービスではセキュリティが最重要視されています。仕様では、Google の KMS から KEK をエクスポートすることはできません。これらの鍵を使用する暗号化と復号化はすべて KMS 内で行う必要があります。これにより、漏えいや誤使用を防ぐことができ、また鍵の使用時に KMS が監査証跡を生成できるようになります。

KMS は一定の時間間隔で自動的に KEK をローテーションし、Google の共通暗号ライブラリを使用して新しい鍵を生成することができます。鍵と言えば 1 つの鍵を指すことが多いですが、厳密にはデータは 1 組の鍵のセットで保護されています。つまり、暗号化にはアクティブな 1 つの

仕様により、KEK を Google の KMS からエクスポートすることはできません。KEK を使用する暗号化と復号化はすべて KMS 内で行う必要があります。これにより、漏えいや悪用を防いでいます。

[4] 以前は AES128 でした。また、これらの鍵の一部はデータ復号化のためアクティブのままとなります。

鍵、復号化には複数の履歴鍵を使用します。履歴鍵の数は鍵のローテーションのスケジュールで決まります。実際の KEK のローテーションのスケジュールはサービスによって異なりますが、標準的なローテーション期間は 90 日です。Google Cloud Storage では KEK を 90 日ごとにローテーションさせています。また、最大で 20 のバージョンを保存できますが、データを少なくとも 5 年に一度再暗号化する必要があります（実際にはデータの再暗号化はもっと短い間隔で行われています）。KMS で保存されている鍵は障害復旧のためバックアップされます。この鍵は無期限で復旧できます。

KEK の使用は、鍵ごとに KMS のアクセス制御リスト（ACL）により管理され、鍵ごとのポリシーが適用されます。承認された Google サービスとユーザーのみが鍵にアクセスできます。各鍵の使用状況は、鍵が必要な個々の操作ごとに追跡されます。つまり、ユーザーが鍵を使用すると、認証され、ログに記録されます。Google 全体のセキュリティ ポリシーとプライバシー ポリシーの一環として、ユーザーによるデータへのアクセスはすべて監査可能です。

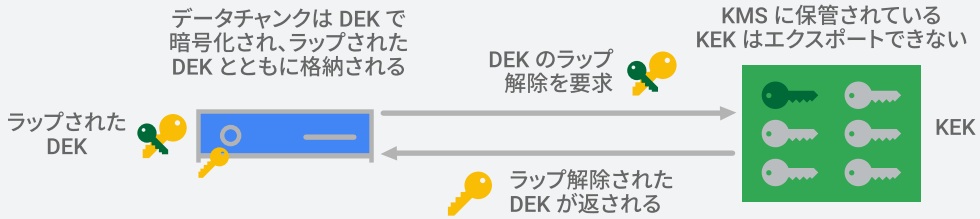


図 3:

ストレージ サービスはデータチャンクの復号化のため、Google の鍵管理サービス（KMS）を呼び出し、そのデータチャンク用のラップ解除されたデータ暗号鍵（DEK）を取得します。

Google Cloud Platform サービスがデータの暗号化済みチャンクにアクセスすると、次の処理が実行されます。

1. サービスがストレージ システムに対して必要なデータを要求します。

2. ストレージ システムはデータが保存されているチャンク（チャンク ID）と、そのチャンクが保存されている場所を特定します。

3. ストレージ システムはチャンクごとに、そのチャンクと一緒に保存されているラップ済み DEK を取得し（この処理はサービスにより実行される場合もあります）、ラップ解除のため KMS に送信します。

4. ストレージ システムは特定されたジョブがデータチャンクにアクセス可能であるかどうかをジョブ識別子とチャンク ID に基づいて確認します。その後、KMS はストレージ システムがサービスに関連付けられた KEK の使用と当該 DEK のラップ解除を承認されているか確認します。

5. KMS は次のいずれかの処理を実行します。
   - ラップ解除された DEK をストレージ システムに返します。これにより、データチャンクが復号化され、サービスに渡されます。また、まれに次の処理が実行されます。

   - ラップ解除された DEK をサービスに渡します。ストレージ システムは暗号化されたデータチャンクをサービスに渡し、サービスはデータチャンクを復号化して使用します。

このプロセスはローカル SSD などの専用ストレージ デバイスでは異なり、デバイスがデバイスレベルの DEK の管理と保護を行っています。

暗号鍵の階層と信頼のルート

Google の KMS は「KMS マスター鍵」と呼ばれるルート鍵で保護されています。この鍵は KMS のすべての KEK をラップします。この KMS マスター鍵は AES256[5] で、これ自体が「ルート KMS」と呼ばれる別の鍵管理サービスに保存されています。ルート KMS にははるかに少ない数（12 個程度）の鍵が保存されています。セキュリティ向上のため、ルート KMS は一般的な本番機では実行されず、各 Google データセンターの専用機でのみ実行されます。

ルート KMS にも専用のルート鍵があり、「ルート KMS マスター鍵」と呼ばれています。この鍵も AES256[6] で、ピアツーピア インフラストラクチャに保存されています。このインフラストラクチャは「ルート KMS マスター鍵ディストリビュータ」と呼ばれていて、これらの鍵をグローバルに複製します。ルート KMS マスター鍵ディストリビュータはルート KMS と同じ専用機の RAM にある鍵のみを保持し、使用が適切であるかログにより確認します。ルート KMS のインスタンスごとに、ルート KMS マスター鍵ディストリビュータのインスタンスが 1 つ実行されます（ルート KMS マスター鍵ディストリビュータの導入はまだ段階的ですが、同様の方式でピアツーピアではないシステムが順次置き換えられています）。

Google の信頼のルートであるルート KMS マスター鍵は RAM に保持されますが、グローバルに再起動する場合に備えて、少数の Google のロケーションにある物理的に安全な場所にも保管されます。

[5] 以前は AES128 でした。これらの鍵の一部はデータ復号化のためアクティブのままとなります。
[6] 以前は AES128 でした。これらの鍵の一部はデータ復号化のためアクティブのままとなります。

ルート KMS マスター鍵ディストリビュータの新しいインスタンスが実行されると、すでにディストリビュータのインスタンスを実行しているホストの名前のリストが設定されます。ディストリビュータのインスタンスには実行中の他のインスタンスからのルート KMS マスター鍵を格納できます。後述の障害復旧メカニズムの場合を除き、ルート KMS マスター鍵は専用のセキュリティが設定された少数の RAM にしかありません。

ルート KMS マスター鍵ディストリビュータのすべてのインスタンスが同時に再起動する場合に対応できるように、Google の物理的に離れた 2 か所のグローバル ロケーションにある物理的に安全な場所に設置された安全なハードウェア デバイスにもルート KMS マスター鍵がバックアップされます。このバックアップが必要となるのは、ディストリビュータのすべてのインスタンスが同時に停止した場合（全世界で一斉に再起動する場合など）に限られます。これらの格納場所に立ち入ることができるのは、20 名に満たない Google 社員のみです。

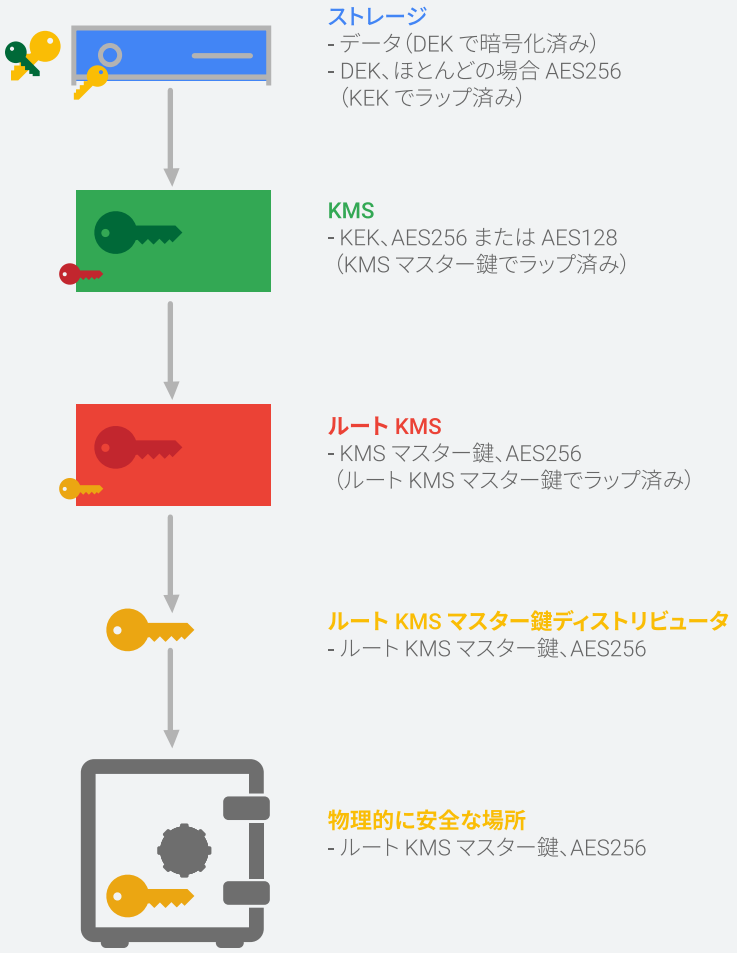


図 4:

暗号鍵階層は DEK によりデータチャンクを保護しています。DEK は KMS の KEK でラップされ、KMS はルート KMS とルート KMS マスター鍵ディストリビュータにより保護されています。

まとめ

- データはチャンク化され、DEK で暗号化されます。
- DEK は KEK により暗号化されます。
- KEK は KMS に保存されます。
- KMS は全世界のデータセンターにある複数のマシンで実行されます。
    - KMS 鍵は、ルート KMS に保存された KMS マスター鍵でラップされます。
- ルート KMS は KMS よりはるかに規模が小さく、各データセンターの専用機でしか実行されません。
    - ルート KMS 鍵は、ルート KMS マスター鍵ディストリビュータに保存されているルート KMS マスター鍵によりラップされます。
- ルート KMS マスター鍵ディストリビュータはグローバルに設置されている専用機の RAM で同時に実行されるピアツーピア インフラストラクチャです。それぞれが、実行中の他のインスタンスから鍵材料を取得します。
    - ディストリビュータのすべてのインスタンスが停止（全体シャットダウン）すると、マスター鍵はごく少数の Google 拠点の（物理的な）格納場所にある（別の）安全なハードウェアに保存されます。
- ルート KMS マスター鍵ディストリビュータの導入はまだ進行中ですが、同様の方式でピアツーピアではないシステムに順次置き換えられています。

グローバルな可用性とレプリケーション

高可用性と低レイテンシ、鍵へのグローバルなアクセスはどのレベルでも非常に重要です。鍵管理サービスを Google 全体で使用できるようにするには、これらの特性が必須となります。

このため、KMS はスケーラブルなものになっており、世界中の Google のデータセンターで何千回も複製されています。KMS は Google の本番環境で使用されている一般的なマシンで実行されます。KMS のインスタンスはグローバルで実行され、Google Cloud Platform のオペレーションをサポートします。その結果、どの鍵オペレーションもレイテンシが非常に低くなります。

ルート KMS は各データセンターにあるセキュリティ オペレーション専用の複数のマシンで実行されます。ルート KMS マスター鍵ディストリビュータはこれらのマシンで実行され、ルート KMS と 1 対 1 の関係にあります。ルート KMS マスター鍵ディストリビュータにより、ゴシッピング プロトコルを使用した配布メカニズムが実現されています。ディストリ

ビュータの各インスタンスは、一定の時間間隔でランダムに他のインスタンスを選択して鍵を比較し、鍵のバージョンの差異を一致させます。このモデルにより、Google のすべてのインフラストラクチャが 1 つのノードに依存することはなくなり、Google は高可用性を確保しながら鍵材料の維持と保護を行うことができます。

Google の共通暗号ライブラリ

Google では「Keyczar[7]」という共通暗号ライブラリを使用しています。このライブラリにより、BoringSSL[8] を使用する暗号アルゴリズムが実装されます。Keyczar はすべての Google デベロッパーが利用できます。この共通ライブラリは広範囲からアクセスできるため、少人数の暗号作成者チームが 1 チームあれば、この厳密に管理および審査されたコードを実装するには十分です。Google のすべてのチームが自分で暗号を作成する必要はありません。専門の Google セキュリティ チームが、すべてのプロダクトを対象にこの共通暗号ライブラリの維持を行います。

Keyczar 暗号ライブラリは数多くの暗号鍵の種類やモードに対応しています。これらの暗号鍵は定期的に審査され、最新の攻撃に対応できるようになっています。

このドキュメントの発行時点においては、Google は次の暗号化アルゴリズムを使用して、DEK と KEK の保存時の暗号化を行っています。Google の機能やセキュリティの強化に伴い、これらのアルゴリズムは変更される場合があります。

> Google では、少人数の暗号作成者チームが共通の暗号ライブラリを厳密に管理、制御、審査し、広範囲からこの暗号ライブラリにアクセスできるようにしているため、Google のすべてのチームが各自で暗号を作成する必要がありません。

基本暗号	推奨されるプロトコル	サポートされるその他のプロトコル[9]
対称暗号化	• AES-GCM（256 ビット）	• AES-CBC および AES-CTR（128 ビットと 256 ビット） • AES-EAX（128 ビットと 256 ビット）
対称署名（認証のため上記の AES-CBC および AES-CTR とあわせて使用されている場合）	• HMAC-SHA256	• HMAC-SHA512 • HMAC-SHA1

[7] Keyczar の古いバージョンの 1 つがオープンソース化されていますが、このオープンソースバージョンは最近は更新されていないため、内部開発が反映されていません。
[8] OpenSSL も Google Cloud Storage の一部で使用されています。
[9] 他の暗号プロトコルもライブラリにあり、以前はサポートされていましたが、このリストは Google Cloud Platform で主に使用されるものを取り上げています。

各 Google Cloud Platform プロダクトでの暗号化の粒度

Google Cloud Platform の各サービスでは、暗号化のためそれぞれ異なる粒度でデータを分割しています。

	Google Cloud Platform サービス	顧客データ暗号化の粒度[10] （1 つの DEK で暗号化されるデータのサイズ）
ストレージ	クラウド ストレージ	データごとのチャンク（通常は 256 KB～8 MB）
	Cloud SQL	• 第 2 世代: インスタンスごと。Google Compute Engine と同様（各インスタンスには複数のデータベースを格納可能） • 第 1 世代: インスタンスごと
	クラウド データベース	データチャンクごと[11]
	クラウドの Bigtable	データチャンクごと（テーブルごとに複数）
コンピューティング	Compute Engine	• ディスクごとに複数 • スナップショット グループごと、スナップショット グループ マスター鍵から派生した個別のスナップショット範囲を使用 • イメージごと
	App Engine[12]	データチャンクごと[13]
	Container Engine	ディスクごとに複数、永続ディスクで使用
	Container Registry	Google Cloud Storage に保存、データチャンクごと
ビッグデータ	BigQuery	データセットごとに複数
	Cloud Dataflow	Google Cloud Storage に保存、データチャンクごと
	Cloud Dataproc	Google Cloud Storage に保存、データチャンクごと
	Cloud Datalab	Cloud Bigtable に保存、ノートブック ファイルごと
	Cloud Pub/Sub	1 時間ごと、最大 100 万件のメッセージ[14]

[10] 顧客コンテンツの暗号化の粒度を指します。これにはリソース名などの顧客メタデータは含まれません。一部のサービスでは、すべてのメタデータが 1 つの DEK で暗号化されます。
[11] 1 顧客について一意であるわけではありません。
[12] アプリケーション コードやアプリケーション設定が含まれます。App Engine で使用されるデータは、顧客の設定により、Cloud Datastore、Cloud SQL または Cloud Storage のいずれかに保存されます。
[13] 1 顧客について一意であるわけではありません。
[14] Cloud Pub/Sub はメッセージの暗号化に使用する DEK を 1 時間ごとにローテーションします。暗号化するメッセージが 100 万件に達する場合はこの間隔が短くなります。 顧客について一意であるわけではありません。

Cloud 顧客向けのその他の暗号化オプション

Google Cloud Platform のデフォルトの暗号化だけでなく、より高度な制御のため追加の暗号化オプションと鍵管理オプションの準備も進めています。

Google Cloud Platform 顧客が次の作業をできるようにすることが目標です。

- データの最終管理者となり、そのデータのアクセスと使用を最高の粒度で制御し、データのセキュリティとプライバシーの両方を確保する。
- クラウドでホストしているデータの暗号化を、現在のオンプレミスと同等、またはそれ以上の水準で管理する。
- リソース全体で、証明可能かつ監査可能な信頼のルートを使用する。
- ACL の使用以外の方法も利用してデータをさらに分離する。

顧客は、Google Cloud Platform で管理している既存の暗号鍵を、ユーザー入力暗号鍵機能により使用することができます。この機能は、ストレージ レイヤの暗号化用として Google Cloud Storage と Google Compute Engine で使用できます。

目下のところ、Google は新しい暗号化オプションの導入に取り組んでいます。利用できる状態になったら、詳細をお知らせする予定です。

Google では、Google Cloud Platform の顧客に追加の暗号化オプションと鍵管理オプションを提供するための準備を進めています。

暗号化の研究とイノベーション

暗号化の進歩に対応するため、Google は世界クラスのセキュリティ エンジニアのチームを組み、暗号化技術の学習、開発、改良を進めています。Google のエンジニアは標準化プロセスと、一般的な暗号化ソフトウェアの維持管理に参加しています。Google では暗号化に関連する研究

の内容を定期的に公開し、業界関係者や一般の人々が Google の知識を活用できるようにしています。たとえば、2014 年には、SSL 3.0 暗号化に重大な脆弱性（POODLE と呼ばれています）があることを発見し、2015 年には OpenSSL にリスクの高い脆弱性があることを突き止めています。

Google はクラウド サービスにおいて業界最先端となるべく計画を進めています。新しい暗号化技術の開発、実装、研究のため、Googleではチームを編成して以下の項目に取り組んでいます。

- 部分的準同型暗号。暗号化された状態のデータに対して一部のオペレーションを行うことができます。これにより、クラウドからはメモリ内のデータであってもプレーンテキスト形式で読み取ることはできません。この技術の活用例として、Google が実験的に導入している暗号化 BigQuery クライアントがあげられます。このサービスは一般公開されています。

- 形式保存暗号。暗号化された状態のデータに対して、一部の比較およびランキング演算を行うことができます。

- ポスト量子暗号。これにより、巧妙な量子攻撃に弱い既存の基礎暗号を、このような攻撃に強いとされるポスト量子暗号に置き換えられるようになります。ここで最も重要なことは、格子ベースの公開鍵暗号の研究とプロトタイプ化です。これには NIST が推奨するポスト量子アルゴリズムも含まれます。格子ベースの暗号は現在、ポスト量子界で利用される可能性が最も高い暗号化技術のひとつとされています。一方で、格子ベースの暗号は、最善のアルゴリズム、具体的なパラメータ、暗号解読という点でまだ歴史が浅いものです。対称鍵暗号化と MAC はこれまでに知られている量子アルゴリズムの登場により弱いものとなっていますが、鍵のサイズを倍にすることにより、量子アルゴリズム界で同等のセキュリティを実現することができます。

Google は
世界規模のセキュリティ
エンジニア チームを
組み、暗号化技術の
学習、開発、改良を
進めています。

その他のリファレンス

Google Cloud Platform のセキュリティ

Google Cloud Platform セキュリティの一般的な情報については、 Google Cloud Platform ウェブサイトのセキュリティを参照してください。

Google Cloud Platform のコンプライアンス

Google Cloud Platform のコンプライアンスとコンプライアンス証明書については、 Google Cloud Platform ウェブサイトのコンプライアンスを参照してください。この項には Google の公開 SOC3 監査レポートも掲載されています。

G Suite のセキュリティ

G Suite の暗号化と鍵管理については、G Suite 暗号化に関するホワイトペーパーを参照してください。このホワイトペーパーの内容はここで紹介しているものとほぼ同じですが、G Suite だけが対象となっています。すべての G Suite ソリューションについて、Google は顧客データの保護に努め、セキュリティ保護の方式についても可能な限り透明性を確保するよう努めています。

G Suite のセキュリティに関する一般的な情報については、『G Suite Security and Compliance Whitepaper』を参照してください。

Google Cloud Platform のデフォルトの暗号化だけでなく、より高度な制御のため追加の暗号化オプションと鍵管理オプションの準備も進めています。

Google Cloud で転送されるデータの暗号化

Google Cloud の暗号化に関するホワイトペーパー

Google Cloud

Google Cloud

目次

Google Cloud

これは、Google が実践している暗号化によるユーザーデータ保護について論じたホワイトペーパーの 3 番目にあたります。他の 2 つは、Google Cloud Platform でのデータの暗号化と G Suite での暗号化です。Google が暗号化をどのように活用しているかについては、これらのドキュメントでも解説しています。このホワイトペーパーでは、Google Cloud Platform と G Suite を含む Google Cloud での転送データの暗号化について詳しく説明します。

すべての Google サービスについて、Google はお客様のデータの厳重な保護に努め、採用しているセキュリティ保護の方式についても可能な限り透明性を確保するよう努めています。

このドキュメントの内容は、2017 年 11 月の時点で正確なものです。このホワイトペーパーは、執筆された時点での現状を取り上げています。Google は継続的にお客様データの保護の強化を進めており、今後、Google Cloud のセキュリティ ポリシーやシステムは変更される可能性があります。

概要

- Google は、転送中のデータの真正性、整合性、プライバシーを確保するため、複数のセキュリティ対策を導入しています。
- Google または Google 代行者が管理していない物理境界の外側へデータを転送するときは、1 つ以上のネットワーク レイヤで暗号化され認証されます。Google または Google 代行者が管理している物理境界の内側で転送されるデータは、通例、認証の対象になりますが、必ずしも暗号化されません。
- 確立されている接続に応じて、転送されるデータにデフォルトの保護設定を適用します。たとえばユーザーと Google Front End（GFE）の間では、TLS を使用して通信を保護しています。
- WAN 経由のデータの暗号化について追加要件を求める Google Cloud のお客様は、データがユーザーからアプリケーション、または仮想マシン（VM）から仮想マシンへと移動する際に追加の保護を実装できます。保護の手段としては、IPSec トンネル、Gmail S/MIME、マネージド SSL 証明書、Istio があります。
- Google は、ユーザーや転送先にかかわらずすべてのデータの暗号化が適用されるよう、関係先と積極的に連携しています。Google は、転送されるデータの暗号化およびインターネットでのデータ セキュリティの利用を広く促進するものとして、Certificate Transparency、 Chrome API、セキュアな SMTP を含め、いくつかのオープンソース プロジェクトを運営しています。
- Google が目指しているのは、転送されるデータの暗号化に関して、業界のリーダーであり続けることです。この目標の達成に向けて、暗号化技術の開発と改善にリソースを投入しています。この領域での成果としては、Key Transparency とポスト量子暗号の分野における技術革新があります。

1. はじめに

パブリック クラウド プロバイダを選定する際、セキュリティが決定要因となることは少なくありません。Google では、セキュリティを最も重要な項目と見なしています。インターネット上で転送されるか、Google のインフラストラクチャ内部で転送されるか、Google のサーバーに保存されるかにかかわらず、お客様のデータの保護に精力的に取り組んでいます。

Google のセキュリティ戦略において柱となっているのは、保存されるデータと転送されるデータの両方の認証、整合性、暗号化です。このホワイトペーパーでは、Google Cloud での転送データの暗号化に関するアプローチを説明します。

保存されるデータについては、Google Cloud Platform でのデータの暗号化をご覧ください。Google セキュリティの全般的な概略については、Google インフラストラクチャのセキュリティ設計の概要をご覧ください。

対象読者: Google Cloud を利用中、または利用を検討中の CISO およびセキュリティ オペレーション チーム。

前提条件: この概要で説明した内容に加え、暗号化と暗号プリミティブについて基本的な知識があること。

転送されるデータの暗号化は、サイトとクラウド プロバイダの間、または 2 つのサービスの間でデータが転送されているとき、通信が傍受された場合にデータを保護します。

1.1 認証、整合性、暗号化

Google は、転送中のデータの真正性、整合性、プライバシーを確保するため、複数のセキュリティ対策を導入しています。

- **認証:** データソース（人間またはプロセス）と宛先を検証します。
- **整合性:** ユーザーからのデータが、改変されることなく宛先に到達することを確認します。
- **暗号化:** データが転送される間、データを判読不能にして機密の状態に維持します。

このホワイトペーパーでは、Google Cloud での暗号化とともに、お客様のデータを暗号化によってどのように保護しているのかに焦点を当てます。暗号化は、読み取り可能なデータ（プレーンテキスト）を判読不能な状態（暗号テキスト）に変換して、データの所有者から承認されている

相手だけが読み取れるようにすることを目的とする処理です。暗号化の処理に使用されるアルゴリズムは公開されているものですが、暗号テキストの復号化で必要になる鍵は非公開です。転送されるデータの暗号化では、多くの場合、楕円曲線に基づく Diffie-Hellman 方式など、非対称暗号鍵交換によって、データの暗号化に使用される共有対称鍵を生成します。暗号化の詳細については、最新の暗号の概要をご覧ください。

暗号化を利用することで、以下の 3 つの状態のデータを保護できます。

- **保存されるデータの暗号化**:保存されるデータを暗号化することで、システムのセキュリティ侵害やデータの不正取得からデータを保護します。保存されるデータの暗号化には、多くの場合、Advanced Encryption Standard（AES）が使用されます。

- **転送されるデータの暗号化**: サイトとクラウド プロバイダ間、または 2 つのサービス間でデータが転送されているときに、通信が傍受された場合にデータを保護します。この保護は、データを転送の前に暗号化し、エンドポイントを認証し、到着時にデータを復号化し、検証することで可能になります。たとえば、転送中のセキュリティを確保するため、転送されるデータには Transport Layer Security (TLS)、メール メッセージには Secure/Multipurpose Internet Mail Extensions（S/MIME）の暗号化方式がよく使われます。

- **使用されるデータの暗号化**: サーバーによる演算の実行に使用されるデータを保護します（準同型暗号化など）。

暗号化は、広範にわたるセキュリティ戦略を構成する要素の 1 つです。転送されるデータの暗号化では、接続が確立され、認証された後、以下の手段によって、潜在的な攻撃者からデータを防護します。

- 信頼性の低いネットワークの下位レイヤ（通常はサードパーティが提供）への依存を軽減する。

- 潜在的な攻撃対象領域を縮小する。

- 通信が傍受された場合に、攻撃者がデータにアクセスすることを阻止する。

適切な認証、整合性、暗号化が確保されていれば、ユーザー、端末、プロセスの間でやり取りされるデータを、悪意のある環境にある場合でも保護できます。以下のセクションでは、転送されるデータの暗号化に対する Google のアプローチおよび暗号化の適用箇所について説明します。

2.Google のネットワーク インフラストラクチャ

2.1 Google のネットワークの物理境界

Google は、Google または Google 代行者が管理している物理境界内から外側へとデータが転送されるとき、転送されるデータにさまざまな保護を適用しています。物理境界とは、Google または Google 代行者が管理している物理空間の防壁であり、そこでは Google が厳格な保護対策を実施できます。これらの場所への立ち入りは制限され、厳重に監視されています。ハードウェアを操作できるのは、Google のごく一部の社員のみです。これらの物理境界の内側でやり取りされるデータは、通例、認証されていますが、デフォルトでは必ずしも暗号化されません。想定される脅威の形態に基づいて、お客様がセキュリティ対策を追加適用できるようにしています。

グローバルなインターネットでは、規模の関係上、Google の WAN 内のファイバー リンクに対して、あるいは、Google または Google 代行者が管理する領域の外側に対して、これらと同様の物理セキュリティ統制を導入することは不可能です。したがって、物理的な信頼境界の外側では、追加的な保護対策を自動的に適用しています。これらの保護対策に含まれているのが、転送されるデータの暗号化です。

2.2 トラフィックのルーティング方法

前のセクションでは、Google のネットワークの物理境界について、また、この境界の外側に送信されるデータに各種保護をどのように適用しているかについて説明しました。転送されるデータの暗号化が Google でどのように機能しているかを十分に理解するには、インターネット上で、トラフィックがどのようにルーティングされているのかについても説明が必要です。このセクションでは、リクエストがエンドユーザーから目的の Google Cloud サービスまたはお客様のアプリケーションへとどのように到達し、サービス間でトラフィックがどのようにルーティングされるのかを説明します。

Google Cloud サービスは、Google がお客様に提供しているモジュール構造のクラウド サービスです。このサービスには、コンピューティング、データ ストレージ、データ分析、機械学習などが含まれています。たとえば、Google Cloud Storage と Gmail はいずれも Google Cloud のサービスです。**お客様のアプリケーション**は、Google Cloud でホストされ、お客様が Google Cloud サービスを利用してビルドし、デプロイすることのできるアプリケーションです。Google Cloud でホストされるお客様の

Google は、Google または Google 代行者が管理している物理境界内から外側へとデータが転送されるとき、転送されるデータにさまざまな保護を適用しています。物理境界とは、Google または Google 代行者が管理している物理空間の防壁であり、そこでは Google が厳格な保護対策を実施できます。

アプリケーションまたはパートナー ソリューションは、Google Cloud サービスとは見なされません[1]。たとえば、Google App Engine や Google Container Engine、あるいは Google Compute Engine の VM を利用してお客様がビルドするアプリケーションは、お客様のアプリケーションです。

以下で説明する 5 種類のルーティング リクエストを図 1 に示します。この図は、各種のネットワーク　コンポーネント間で生じるインタラクションと、個々の接続に適用されるセキュリティを表しています。

エンドユーザー（インターネット）から Google Cloud サービスへの通信

Google Cloud サービスは、Google Front End（GFE）と呼ばれるグローバルな分散型システムを利用して、世界各地からリクエストを受け付けています。GFE は、着信する HTTP(S)、TCP、TLS プロキシ トラフィックの終端として機能し、DDoS 攻撃への対抗策を提供し、Google Cloud サービス本体へのトラフィックのルーティングと負荷分散を行います。ユニキャストまたはエニーキャストを通じてアドバタイズされるルートを備えた GFE の接続拠点は、世界各地に存在しています。

GFE は、Google Cloud サービスへのトラフィックをプロキシしています。ユーザーからのリクエストを Google のネットワーク バックボーン経由で Google Cloud サービスにルーティングします。この接続は、GFE を送信元、Google Cloud サービスまたはお客様のアプリケーション フロントエンドを宛先として、Google または Google 代行者が管理する物理境界の外側への通信が生じる時点で、認証され、暗号化されます。図 1 に、該当するインタラクションを示しています（接続 A）。

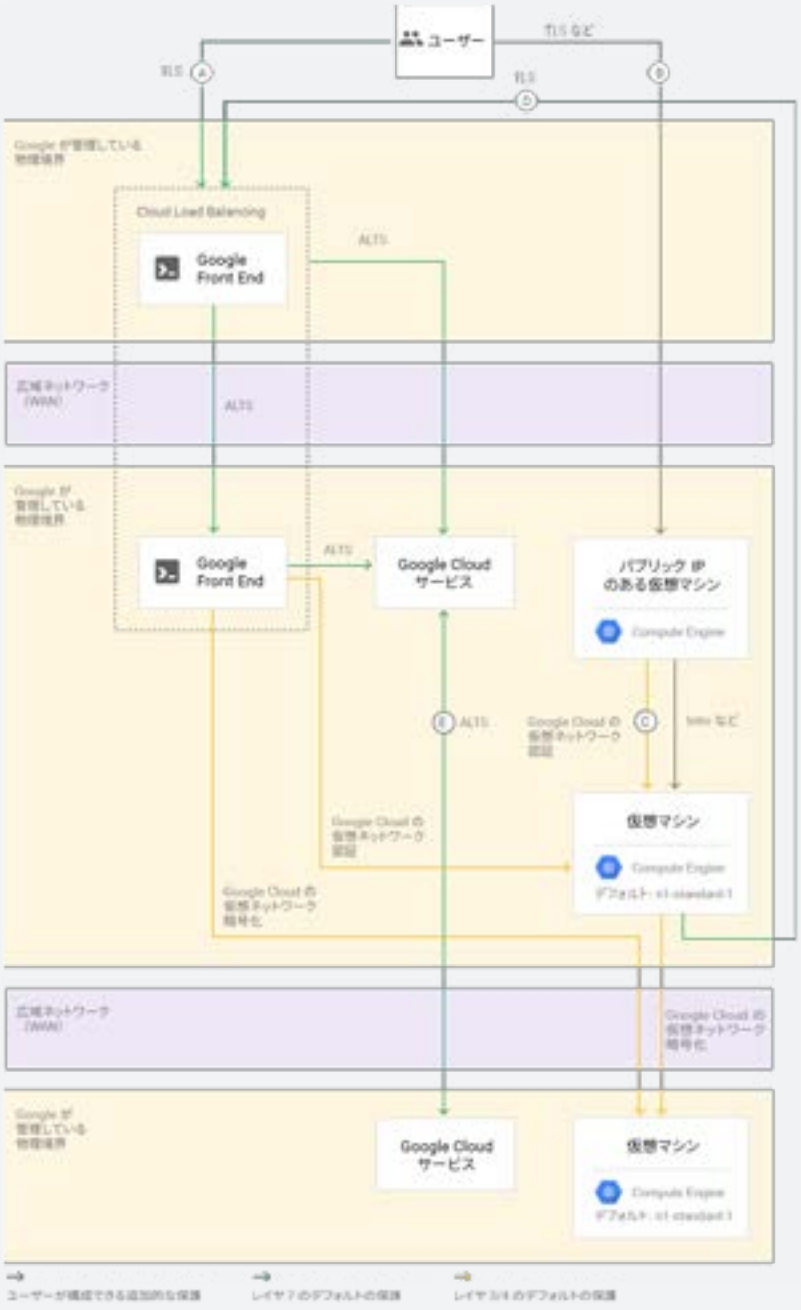
エンドユーザー（インターネット）から、Google Cloud でホストされているお客様のアプリケーションへの通信

Google Cloud でホストされているお客様のアプリケーションに、インターネットからのトラフィックをルーティングできるようにするには、いくつかの方法があります。トラフィックのルーティング方法は、以下で説明するとおり、構成によって異なります。図 1 に、該当するインタラクションを示しています（接続 B）。

- **Google Cloud HTTP(S) または TCP/SSL プロキシ Load Balancer による外部ロードバランサを使用:** Google Compute Engine 上の VM でホストされているお客様のアプリケーションでは、Google Cloud Load Balan-cer（GCLB）サービスを使用て、HTTP(S)、TLS、TCP　の接続を終端処理し、そのトラフィックを VM にプロキシ、ルーティング、分散

Google Cloud サービスは、Google Front End（GFE）と呼ばれるグローバルな分散型システムを利用して、世界各地からリクエストを受け付けています。GFE は、着信する HTTP(S)、TCP、TLS プロキシ トラフィックの終端として機能し、DDoS 攻撃への対抗策を提供し、Google Cloud サービス本体へのトラフィックのルーティングと負荷分散を行います。

[1] パートナー ソリューションには、Cloud Launcher で提供されるソリューションと、Cloud Dataprep のようなパートナーとの共同開発によるプロダクトが含まれます。



[図 1]

Google のネットワークにおけるデフォルトの保護とオプションの保護

できます。これらのロードバランサ サービスは GFE によって実装され、Google Cloud サービスへのトラフィックと同様に GFE がトラフィックの終端やルーティングを行います。GCLB が GFE の間でトラフィックをルーティングするとき、Google または Google 代行者が管理する物理境界からトラフィックが離れる時点で、接続は認証され、

暗号化されます。 GCLB が、GFE と、お客様の VM をホストしている物理マシンとの間でトラフィックをルーティングするときは、Google または Google 代行者が管理する物理境界をトラフィックが離れる時点で、このトラフィックは認証され、暗号化されます。

- **Google Cloud HTTP(S) または TCP/SSL プロキシ Load Balancer による外部ロードバランサを使用**: Google Compute Engine 上の VM でホストされているお客様のアプリケーションでは、Google Cloud Load Balancer (GCLB) サービスを使用して、HTTP(S)、TLS、TCP の接続を終端処理し、そのトラフィックを VM にプロキシ、ルーティング、分散できます。これらのロードバランサ サービスは GFE によって実装され、Google Cloud サービスへのトラフィックと同様に GFE がトラフィックの終端やルーティングを行います。GCLB が GFE の間でトラフィックをルーティングするとき、Google または Google 代行者が管理する物理境界からトラフィックが離れる時点で、接続は認証され、暗号化されます。 GCLB が、GFE と、お客様の VM をホストしている物理マシンとの間でトラフィックをルーティングするときは、Google または Google 代行者が管理する物理境界をトラフィックが離れる時点で、このトラフィックは認証され、暗号化されます。

  HTTPS ロードバランサの場合、エンドユーザーと GFE の間の接続は、お客様からロードバランサ用に提供された証明書を使用して、TLS または QUIC で暗号化され、認証されます。HTTP ロードバランサの場合、エンドユーザーと GFE の間の接続は暗号化されず、認証されません。

  SSL ロードバランサの場合、エンドユーザーと GFE の間の接続は、同様にお客様から提供された証明書を使用して、TLS で暗号化されます。 TCP ロードバランサの場合、エンドユーザーと GFE の間は暗号化されません。ただし、お客様のアプリケーションで、エンドユーザーと VM の間を独自に暗号化できます。

- **外部 IP またはネットワーク ロードバランサ IP を利用して、VM への直接接続を使用:** VM の外部 IP またはネットワーク ロードバランサ IP を通じて接続する場合、その接続は GFE を経由しません。この接続はデフォルトでは暗号化されず、セキュリティは、ユーザーの裁量で提供されるものになります。

- **Cloud VPN を使用:** オンプレミスのホストから Google Cloud VM に VPN で接続する場合、その接続は、オンプレミスのホストを始点または終点として、オンプレミスの VPN、Google VPN、Google Cloud VM まで到達し、GFE を経由しません。オンプレミス VPN から Google VPN までは、IPSec で保護されます。Google VPN から

Google Cloud VM への接続は、Google または Google 代行者が管理する物理境界の外側へと通信が向かう時点で、認証され、暗号化されます。

- **Cloud Dedicated Interconnect を使用:** Dedicated Interconnect を使用して接続する場合、その接続は、オンプレミスのホストを直接の始点または終点として、GFE を経由しません。この接続はデフォルトでは暗号化されず、セキュリティは、ユーザーの裁量で提供されるものになります。

仮想マシン間

Google のネットワーク バックボーンで、RFC1918 のプライベート IP アドレスを使用して実行される VM 間ルーティングでは、Google または Google 代行者が管理する物理境界から、その外側へと必要に応じてトラフィックをルーティングすることがあります。VM 間のルーティングとしては、以下の例が挙げられます。

- Compute Engine の VM 同士がリクエストを相互に送信する
- お客様の VM から Cloud SQL など Google が管理する VM に接続する

VM 間の接続は、トラフィックが物理境界の外側に離れる場合、物理境界の内側で暗号化され、認証されます。パブリック IP アドレスを使用する VM 間トラフィックは、デフォルトでは暗号化されず、セキュリティはユーザーの裁量で提供されます。図 1 に、該当するインタラクションを示しています（接続 C）。

仮想マシンから Google Cloud サービスへの通信

VM がリクエストを Google Cloud サービスにルーティングする場合、リクエストは GFE 宛てにルーティングされます（前述のとおり、Google が管理する VM で Google Cloud サービスが実行されている場合は除きます）。GFE は、リクエストを受け取ると、インターネットから着信するリクエストと同様の方法でルーティングします。VM から Google Cloud サービスへのトラフィックの場合は、プライベート Google パスを経由して、GFE の同一のパブリック IP へとルーティングされます。プライベート Google アクセスを利用すると、パブリック IP のない VM から、Google App Engine でホストされている一部の Google Cloud サービスおよびお客様のアプリケーションにアクセスできます（VM の接続先が、Google Compute Engine または Google Container Engine でホストされているお客様のアプリケーションである場合、そのトラフィックは、インターネットから着信するリクエストと同じように、外部パスを経由してルー

ティングされます）。図 1 に、該当するインタラクションを示しています（接続 D）。この種のルーティング リクエストの例としては、Compute Engine VM から Google Cloud Storage へ、または機械学習 API へのリクエストがあります。Google Cloud サービスは、これらの接続を TLS で保護する機能をデフォルトでサポートしています[2]。この保護は、VM から GFE への接続に適用されます。GFE からサービスまでの接続は、物理境界の外側へと離れるものである場合、認証され、暗号化されます。

Google Cloud サービス間

本番環境サービス間のルーティングは、Google のネットワーク バックボーンで発生し、トラフィックは、Google または Google 代行者が管理する物理境界の外側へと必要に応じてルーティングされることがあります。図 1 に、該当するインタラクションを示しています（接続 E）。この種のトラフィックの例としては、Google Cloud Functions をトリガーする Google Cloud Storage イベントがあります。本番環境サービス間の接続は、トラフィックが物理境界の外側に離れる場合、暗号化され、物理境界の内側で認証されます。

3. 転送されるデータのデフォルトの暗号化

Google は、転送されるデータを対象として、デフォルトで適用されるものとユーザーが構成できるものの両方で、さまざまな暗号化の手法を利用しています。利用される暗号化の種類は、OSI レイヤ、サービスの種類、インフラストラクチャの物理コンポーネントによって異なります。以下の図 2 および 3 は、Google Cloud がレイヤ 3、4、7 で適用しているオプションの保護とデフォルトの保護を図解したものです。

このセクションの残りの部分では、転送されるデータを保護するために Google が利用しているデフォルトの保護機能について説明します。

3.1 ユーザーから Google Front End への通信の暗号化

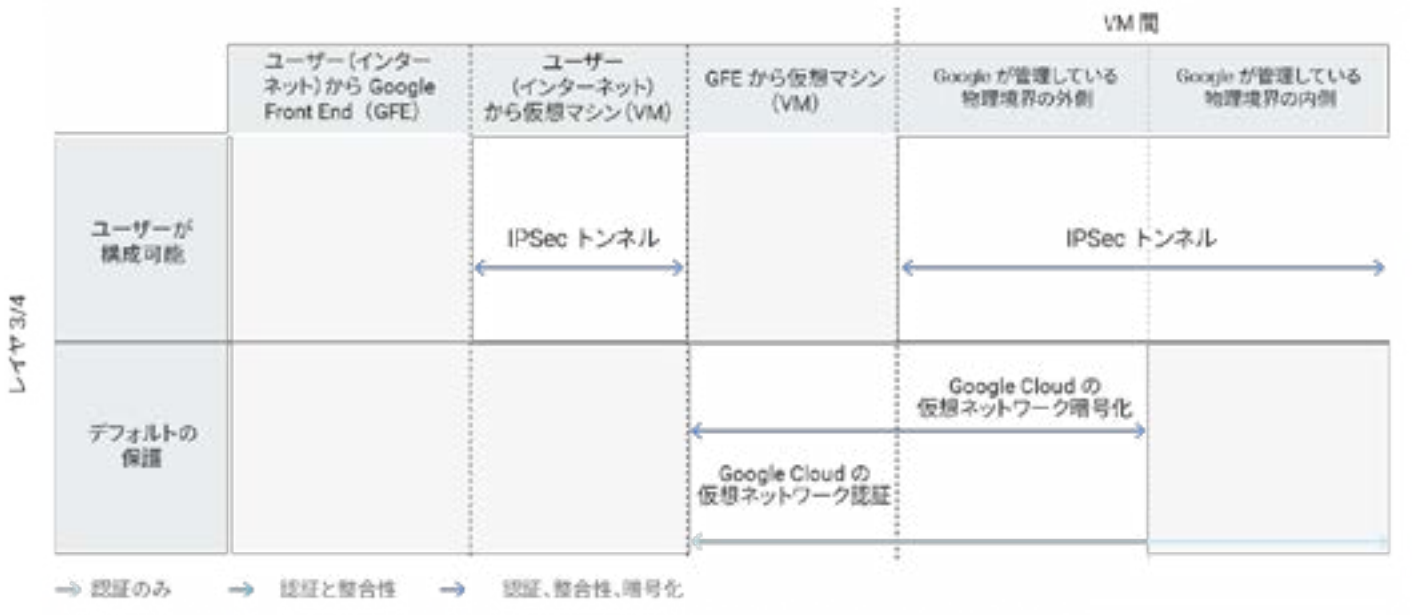
今日では、インターネットでの通信に、多くのシステムが HTTPS プロトコルを使用しています。HTTPS は、TLS 接続上でプロトコルを処理することによってセキュリティを適用し、リクエストとレスポンスの真正性、整合性、プライバシーを保証するものです。HTTPS リクエストを受け付けるため、受信者は、認証局（CA）から発行される公開鍵 / 秘密鍵のペアと X.509 証明書を、サーバーの認証用に要求します。鍵ペアと証明書は、リクエストの送信先ドメイン名を受信者が所有していると証明することによって、ユーザーのリクエストをアプリケーション レイヤ（レイヤ 7）で保護する目的で利用されます。以下のサブセクションでは、ユーザーから GFE までの暗号化を構成している要素について説明します（TLS、BoringSSL、Google の認証局）。これまでに説明したとおり、必ずしも、

[2] この暗号化は無効にすることもできます（たとえば、Google Cloud Storage バケットへの HTTP アクセスの場合）。

すべてのお客様のパスが GFE を経由してルーティングされるわけではありません。GFE が使用されるのは、ユーザーから Google Cloud サービスへのトラフィックと、Google Cloud 上でホストされ、Google Cloud Load Balancing を使用しているお客様のアプリケーションへのトラフィックです。

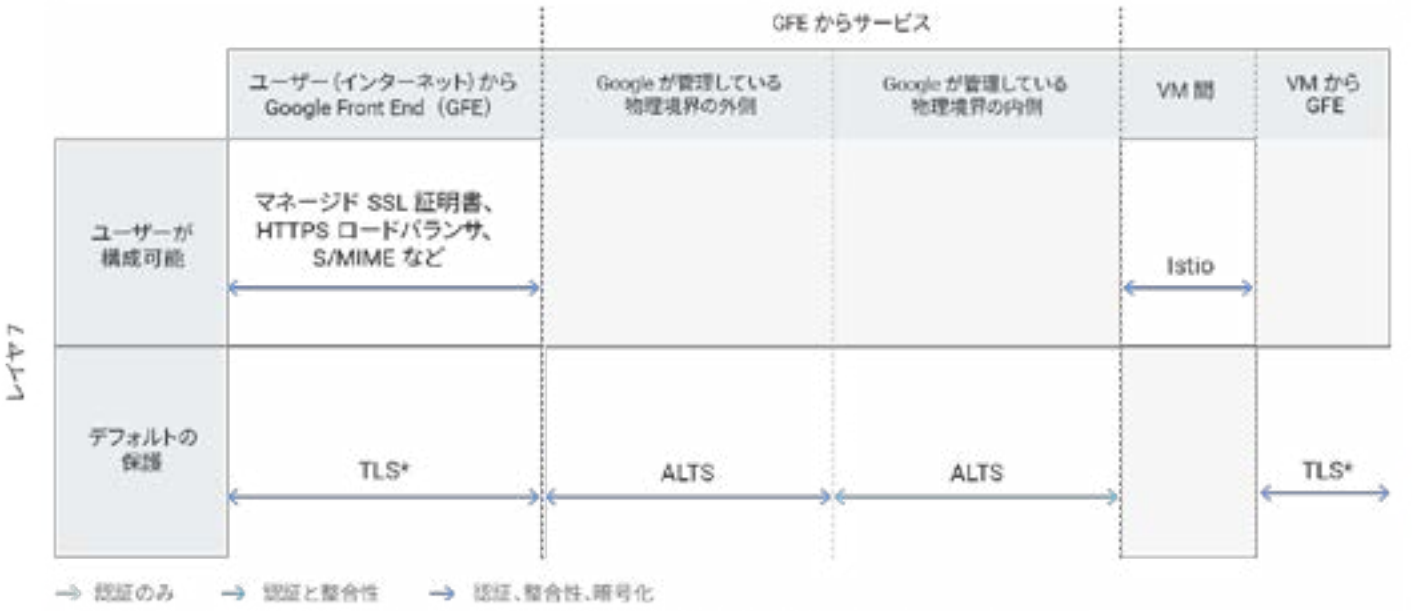
[図 2]

デフォルトでの保護



[図 3]

Google Cloud 全体のレイヤ 7 におけるデフォルトの保護とオプションの保護[3]



* TLS はデフォルトで Google Cloud サービスに適用されます。Google Cloud でホストされているお客様のアプリケーションの場合、これはお客様が構成しなければならない項目です。

[3] レイヤ 7 で保護されていない VM - サービス間の通信も、レイヤ 3 とレイヤ 4 では保護されます。

3.1.1 Transport Layer Security (TLS)

ユーザーが Google Cloud サービスにリクエストを送信するとき、Google は、転送されるデータを保護し、ウェブ（パブリック）認証局発行の証明書を使用して、HTTPS プロトコルによって認証、整合性、暗号化を提供します。ユーザーが GFE に送信するデータは、転送時に Transport Layer Security（TLS）または QUIC で暗号化されます。GFE は、クライアントがどのプロトコルをサポートできるかに応じて、個別の暗号化プロトコルをクライアントとネゴシエートします。可能な場合は、より新しい暗号化プロトコルをネゴシエートします。

GFE の段階的な TLS 暗号化は、エンドユーザーと Google とのインタラクションに適用されるだけでなく、TLS を介した API と Google との（これには Google Cloud も含まれます）インタラクションも促進します。また、TLS 暗号化は、Gmail での外部メールサーバーとのメール交換にも使用されています（詳細については、セクション 4.2.2 をご覧ください）。

Google は、TLS の採用とその実装の強化に関して、業界を先導する存在となっています。この目的を達成するため、Google は、デフォルトで TLS の数多くのセキュリティ機能を使用しています。たとえば、2011 年より TLS 実装の前方秘匿性を利用しています。前方秘匿性とは、あるメッセージを傍受して読み取った攻撃者がその前のメッセージを読み取れないよう、接続を保護する鍵を更新して長く使わないようにする仕組みです。

ユーザーが Google Cloud サービスにリクエストを送信するとき、Google は、転送されるデータを保護し、ウェブ（パブリック）認証局発行の証明書を使用して、HTTPS プロトコルによって認証、整合性、暗号化を提供します。

3.1.2 BoringSSL

BoringSSL は、OpenSSL からフォークされ、Google が管理しているオープンソースの TLS プロトコル実装であり、インターフェースの大部分は OpenSSL と互換性があります。Google が BoringSSL を OpenSSL からフォークしたのは、社内利用と、Chromium および Android のオープンソース プロジェクトのサポートの強化目的で OpenSSL を簡素化するためです。BoringSSL の中核である BoringCrypto は、検証で FIPS 140-2 のレベル 1 相当であるとされています。

GFE の TLS は、BoringSSL を使用して実装されています。表 1 に、クライアントとの通信時に GFE がサポートしている暗号化プロトコルを示します。

[表 1]

Google Front End で Google Cloud サービス用に、BoringSSL 暗号ライブラリで実装される暗号化

プロトコル	認証	鍵交換	暗号化	ハッシュ関数
TLS 1.3[4] TLS 1.2 TLS 1.1 TLS 1.0[5] QUIC[6]	RSA 2048 ECDSA P-256	Curve25519 P-256 (NIST secp256r1)	AES-128-GCM AES-256-GCM AES-128-CBC AES-256-CBC ChaCha20-Poly1305 3DES[7]	SHA384 SHA256 SHA1[8] MD5[9]

3.1.3 Google の認証局

TLS の一環として、サーバーは、接続リクエストを受け取ったときに自身の ID をユーザーに対して証明しなければなりません。この ID 検証は TLS プロトコルで行われ、サーバーは自己を証明する ID が含まれた証明書を提示します。この証明書には、サーバーの DNS ホスト名と公開鍵が収められています。提示された証明書は、接続リクエスト元のユーザーが信頼する発行元認証局（CA）によって署名されています[10]。結果として、サーバーへの接続をリクエストするユーザーは、ルート CA を信頼するだけで済みます。サーバーが、場所を問わずアクセスの対象となる場合、ルート CA は、世界各地のクライアント端末にとって既知のものでなければなりません。今日、大多数のブラウザやその他の TLS クライアント実装は、自身の「ルートストア」において、独自の組み合わせのルート CA を信頼済みとして構成しています。

[4] TLS 1.3 はまだ最終承認されていません。ドラフト版は、Gmail などの特定の Google ドメインに限り、テストを目的として実装されています。
[5] Google では、TLS 1.0 を引き続き使用しているブラウザについては、このプロトコル バージョンをサポートしています。クレジット カード情報を処理する Google サイトでは、Payment Card Industry（PCI）コンプライアンスで TLS 1.0 の使用中止が義務付けられている 2018 年 7 月までに、TLS 1.0 のサポートを終了する予定です。
[6] QUIC の詳細については、https://www.chromium.org/quic をご覧ください。
[7,8,9] 一部のレガシー オペレーティング システムとの後方互換性を確保するため、Google は 3DES、SHA1、MD5 をサポートしています。
[10] チェーン証明書の場合、CA は推移的に信頼されます。

従来、Google は独自の発行元 CA を運営し、Google ドメイン用の証明書への署名に使用していました。ただし、独自のルート CA は運営していませんでした。現在、Google の CA 証明書は世界各地に分散している複数のルート CA によって相互署名されています。その中には、Symantec（GeoTrust）や以前 GlobalSign が運営していたルート（GS Root R2 と GS Root R4）も含まれます。

2017 年 6 月、Google は、Google が所有するルート CA に移行することを発表しました。各地に分散した、Google ドメイン用およびお客様用の証明書を発行するルート CA を運営することを予定しています。

3.1.3.1 ルート鍵の移行と鍵ローテーション

ルート CA 鍵は変更頻度がさほど高いものではなく、新しいルート CA へと移行するには、すべてのブラウザと端末に当該の証明書の信頼を組み込む必要があることから、時間を要します。このため、Google は独自のルート CA をすでに運営しているものの、独自の CA に移行する間、従来型の端末を考慮した移行期間として、複数の第三者ルート CA を引き続き利用します。

ルート CA を新たに構築するにあたっては、鍵セレモニーが義務付けられます。Google では、正当な権限を持つ 6 人のうち 3 人以上が集合し、金庫に保管されている物理的な鍵を使用することを、セレモニーの必須条件としています。これらの人物は、電磁的な干渉から遮断された専用室に参集し、エアギャップ（ネットワークから物理的に隔離された）状態のハードウェア セキュリティ モジュール（HSM）を使用して、鍵と証明書のセットを生成します。この専用室は、Google データセンター内の安全な場所に設けられています。物理的なセキュリティ対策、カメラ、他の立会人など、追加的な統制手段を通じて、手続きが予定どおり実施されるよう万全を期しています。セレモニーが適切に実施された場合、生成される証明書は、発行元名、公開鍵、署名を除いて、サンプル証明書と完全に同一になります。完成したルート CA 証明書は、ブラウザと端末のルート プログラムに送信され、組み込まれます。この手続きの目的は、関連付けられる秘密鍵が 10 年以上にわたって利用し得るものとなるよう、鍵のプライバシーとセキュリティの十分な認知を保証することにあります。

これまでに説明したとおり、CA は秘密鍵を使用して証明書に署名し、それらの証明書は、ユーザー セッションの一環として TLS handshake が開始されたとき、ID を証明するものになります。サーバー証明書に署名するのはルート CA で、この CA はルート CA と同様のプロセスを経て作成されます。中間 CA の証明書は、TLS セッションの一環として配布

されることから、新しい中間 CA への移行は比較的容易です。こうした方法で配布されているため、CA の運営者が、ルート CA の鍵マテリアルをオフラインの状態に維持することも可能になります。

TLS セッションのセキュリティは、サーバーの鍵がどの程度適切に保護されているかに依存します。キーがセキュリティ侵害を受けるリスクをさらに低減するため、Google の TLS 証明書は有効期間が約 3 か月に限定され、およそ 2 週間ごとに証明書がローテーションされます。

サーバーに接続したことのあるクライアントは、秘密チケット鍵[11] を使用することにより、短縮型の TLS handshake を経て前のセッションを再開できることから、これらのチケットは、攻撃者にとってきわめて有用なものになります。Google では、チケット鍵を少なくとも 1 日 1 回ローテーションし、3 日ごとに、すべてのプロパティにわたって鍵を期限切れとしています。セッション鍵チケットのローテーションの詳細については、TLS暗号ショートカットのセキュリティ損害の測定をご覧ください。

3.2 Google Front End からアプリケーションのフロントエンドへの通信

セクション 2.2 で説明したとおり、ユーザーが接続する GFE と、目的のサービスやそれに関連するアプリケーション フロントエンドが別の物理境界に存在する場合があります。この場合、ユーザーのリクエストおよび HTTP をはじめとするその他のレイヤ 7 プロトコルは、TLS で保護されるか、セクション 3.4 で説明する Application Layer Transport Security（ALTS）の保護対象であるリモート プロシージャ コール（RPC）内にカプセル化されてから、セクション 3.4 で説明する Application Layer Transport Security（ALTS）によって保護されます。これらのリモート プロシージャ コールは、認証と暗号化の対象になります。

3.3 Google Cloud の仮想ネットワークの暗号化と認証

Google Cloud の仮想ネットワーク インフラストラクチャでは、トラフィックが Google の物理境界の外側へと離れる時点で暗号化できます。暗号化はネットワーク レイヤで行われ、同一の Virtual Private Cloud（VPC）内の、またはピア VPC ネットワークをまたがるプライベート IP トラフィックに適用されます。

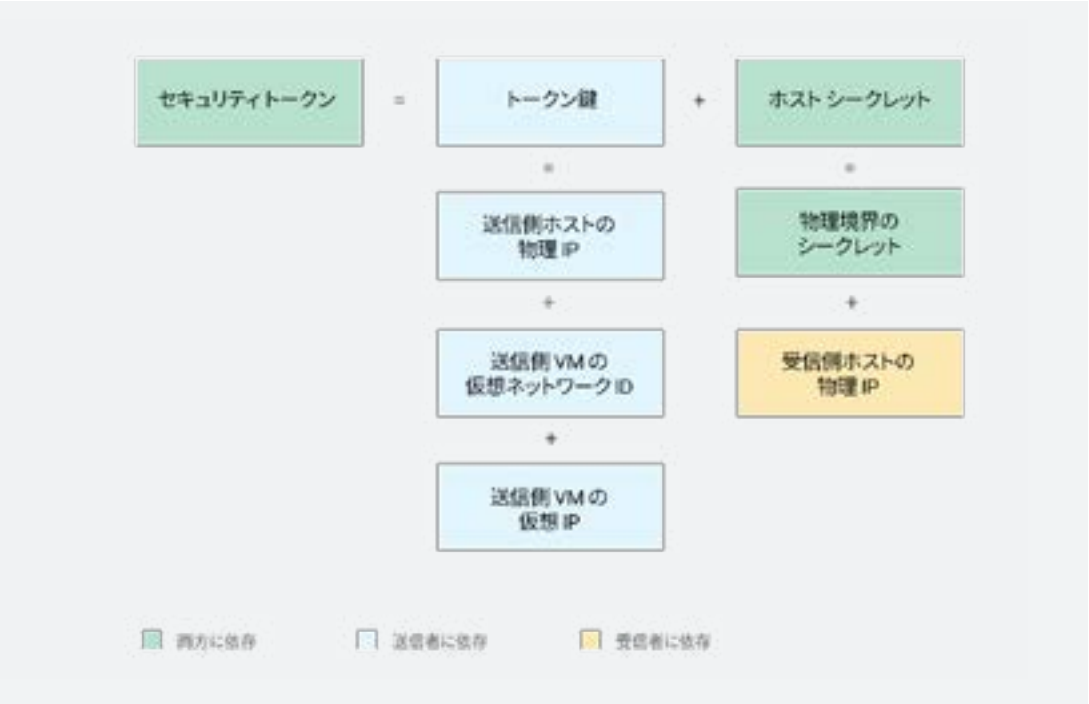
Google は、Google または Google 代行者が管理していない物理境界を横断するネットワークについて、スヌープ、インジェクション、物理配線上でのトラフィック改変を実行し得る活動中の攻撃者から、セキュリティ侵害を受ける可能性があることを前提としています。Google が管理していない、物理境界の外側へとデータが移動するときは、暗号化によって通信の整合性とプライバシーを確保しています。

Google Cloud の仮想ネットワーク インフラストラクチャでは、トラフィックが Google の物理境界の外側へと離れる時点で暗号化できます。暗号化はネットワーク レイヤで行われます。

[11] これは、セッション チケット（RFC 5077）またはセッション ID（RFC 5246）になります。

Google Cloud サービスの場合、リモート プロシージャ コールはデフォルトで ALTS によって保護されます。Google Cloud でホストされるお客様のアプリケーションについては、トラフィックが GFE 経由でルーティングされる場合、たとえば GCLB を使用している場合は、次のセクションで説明する Google Cloud の仮想ネットワーク暗号化によって、VM へのトラフィックが保護されます。

Google は、Galois/Counter Mode（GCM）の Advanced Encryption Standard（AES）を 128 ビット鍵（AES-128-GCM）で使用して、ネットワーク レイヤで暗号化を実装しています。通信ホストの各ペアは、通信の認証と暗号化のためにALTS で保護される制御チャネル上でセッション鍵を確立します。セッション鍵は、それらのホストの間で発生するすべての VM 間通信を暗号化するのに使用され、定期的にローテーションされます。



[図 4]

セキュリティ トークン

ネットワーク レイヤ（レイヤ 3）では、VM 間で発生するすべてのトラフィックを Google Cloud の仮想ネットワークが認証します。セキュリティ トークンにより実行されるこの認証は、セキュリティ侵害を受けたホストをネットワーク上のスプーフィング パケットから保護するものです。

認証の実行中、セキュリティ トークンは、送信者と受信者の認証情報を収めているトンネル ヘッダー内にカプセル化されます。送信側のコントロール プレーン[12] がトークンを設定し、受信側のホストが検証します。セキュリティ トークンは、あらゆる個々のフローについて事前生成され、トークン鍵（送信者の情報を収容）とホスト シークレットで構成されています。Google または Google 代行者が管理する物理境界のあらゆる送信者 - 受信者ペアごとに、1 つのシークレットが存在しています。図 4 は、トークン鍵、ホスト シークレット、セキュリティ トークンの生成プロセスを示しています。

物理境界のシークレットは 128 ビットの疑似乱数であり、HMAC-SHA1 を得ることによって、このシークレットからホスト シークレットが導出されます。物理境界のシークレットは、物理境界のペアのネットワーク コントロール プレーン間で発生する handshake によってネゴシエートされ、数時間ごとに再ネゴシエートされます。個々の VM 間認証に使用されるセキュリティ トークンはこれらの入力などをもとに算出され、所与の送信者 - 受信者についてネゴシエートされる HMAC です。

3.4 サービス間の認証、整合性、暗号化

Google インフラストラクチャ内部のアプリケーション レイヤ（レイヤ 7）では、GFE からサービス、サービスからサービスへの Google リモート プロシージャ コールの認証、整合性、暗号化に、Application Layer Transport Security（ALTS）を使用しています。

ALTS では、サービス アカウントを認証に使用します。Google のインフラストラクチャの内部で実行される各サービスは、関連付けられている暗号化された認証情報を使用して、サービス アカウント ID で実行されます。サービスは、リモート プロシージャ コールの送信時、または他のサービスからの受信時に、自身の認証情報を使用して認証を受けます。ALTSは、内部の認証局を使用してこれらの認証情報を検証します。

Google または Google 代行者が管理する物理境界の内側では、ALTS は「認証と整合性」モードとなり、リモート プロシージャ コールについて、認証と整合性の両方を提供します。Google または Google 代行者が管理する物理境界の外側の WAN トラフィックについては、「認証、整合性、プライバシー」モードとなり、インフラストラクチャのリモート プロシージャ コール トラフィックに対して暗号化を自動的に適用します。現在では、Google Cloud サービスを含め、Google サービスに向かうすべてのトラフィックに対して、これらと同一の保護が適用されています。

ALTS は、GFE からアプリケーション フロントエンドに移動するトラフィックを対象として、HTTP など、その他のレイヤ 7 プロトコルをインフラストラクチャのリモート プロシージャ コール メカニズムにカプセル

Google インフラストラクチャ内部のアプリケーション レイヤでは、GFE からサービス、サービスからサービスへの Google リモート プロシージャ コールの認証、整合性、暗号化に、Application Layer Transport Security（ALTS）を使用しています。

[12] **コントロール プレーン**はネットワークの構成要素の 1 つで、シグナリング トラフィックを伝送し、ルーティングを担当します。

化する目的でも使用されます。この保護によって、アプリケーション レイヤが分離され、ネットワーク パスのセキュリティに依存する状態が解消されます。

サービスが Google または Google 代行者が管理する物理境界の内側にある場合も含めて、「認証、整合性、プライバシー」モードの ALTS 通信だけを送受信するように構成することができます。その一例は Google の内部鍵管理サービスで、Google のインフラストラクチャ内に保存されるデータを保護するための暗号鍵を保管し、管理します。

3.4.1 ALTS プロトコル

ALTS は、相互 TLS に似た安全な handshake プロトコルを備えています。ALTS を使用して通信しようとする 2 つのサービスは、機密情報を送信する前に、この handshake プロトコルを使用して認証を実行し、通信パラメータをネゴシエートします。このプロトコルは、以下の 2 ステップのプロセスとなっています。

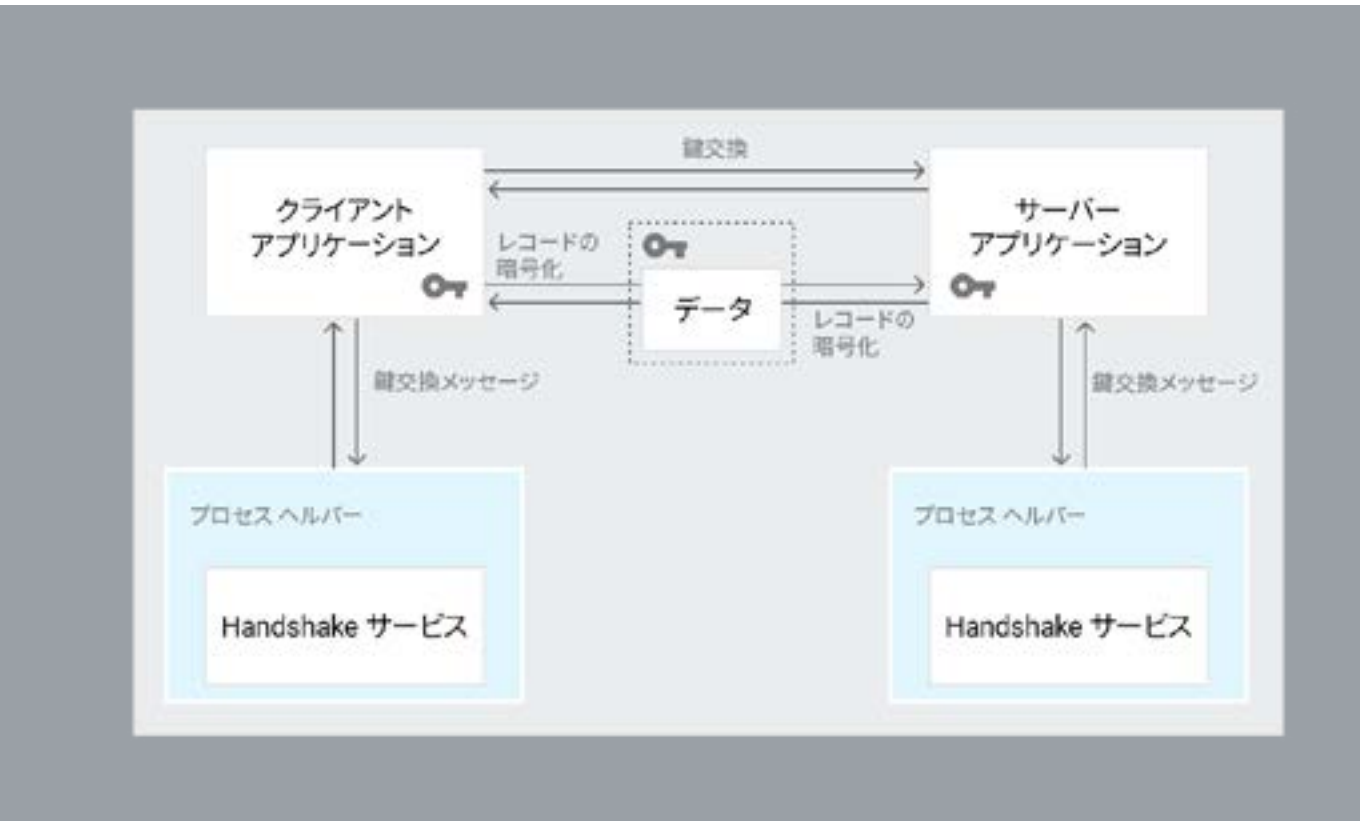
ステップ 1: Handshake

クライアントが、Curve25519 を使用してサーバーとの楕円曲線 Diffie-Hellman（ECDH）handshakeを開始します。クライアントとサーバーは、それぞれ証明済みの ECDH 公開パラメータを証明書に含め、その証明書を Diffie-Hellman 鍵交換に使用します。handshake の結果として、クライアントとサーバーで使用できる共通トラフィック鍵が導出されます。証明書に含まれているピア ID は、アプリケーション レイヤに公開され、承認の可否の決定に使用されます。

ステップ 2: レコードの暗号化

ステップ 1 の共通トラフィック鍵を使用して、クライアントからサーバーへとデータが安全に転送されます。ALTS での暗号化は、BoringSSL をはじめとする暗号化ライブラリを利用して実装されています。整合性は AES-GCM の GMAC によって提供される一方、暗号化は、AES-128-GCM であることが一般的です。

以下の図 5 は、ALTS handshake の詳細を示しています。比較的新しい実装では、handshake を実行するのはプロセス ヘルパーですが、アプリケーションによって直接実行されるケースも依然として存在しています。



[図 5]
ALTS
Handshake

セクション 3.4 の冒頭で説明したとおり、ALTS では認証にサービス アカウントを使用しており、Google のインフラストラクチャで実行される各サービスは、関連付けられている暗号化された認証情報を使用して、サービス ID で実行されます。ALTS handshake の実行中、プロセス ヘルパーは、各クライアント - サーバーペアによって通信で使用される、秘密鍵および対応する証明書にアクセスします。この秘密鍵および対応する証明書 (署名済みのプロトコル バッファ) は、サービスのサービス アカウント ID に対してプロビジョニングされています。

ALTS 証明書
ALTS 証明書には、以下の種類があります。

- **マシン証明書:** 個別のマシンのコアサービスに ID を提供します。約 6 時間ごとにローテーションされます。

- **ユーザー証明書:** コードを作成する Google エンジニア向けのエンドユーザー ID を提供します。約 20 時間ごとにローテーションされます。
- **Borg ジョブ証明書:** Google のインフラストラクチャ内部で実行されるジョブに ID を提供します。約 48 時間ごとにローテーションされます。

ルート証明書署名鍵は Google の内部 CA に保存されます。この CA は Google の外部 CA とは無関係であり、独立しています。

3.4.2 ALTS での暗号化

ALTSでの暗号化は、使用するマシンに応じて、さまざまなアルゴリズムを使用して実装できます。たとえば、大多数のサービスが使用しているのは AES-128-GCM[13] です。ALTS での暗号化については、表 2 でもう少し詳しく説明します。

[表 2]

ALTS での暗号化

マシン	使用されるメッセージ暗号化方法	
最も一般的	AES-128-GCM	
Sandy Bridge 以前	AES-128-VCM	GMAC の代わりに VMAC を使用しており、こうした古いマシンでは効率が若干向上します。

大多数の Google サービスは ALTS か ALTS によって保護される RPC を使用しています。ALTS が使用されない場合は、その他の保護手段が適用されます。以下に例を示します。

- 低レベルのマシン管理サービスとブートストラップ サービスの一部は、SSH を使用
- 低レベルのインフラストラクチャ ロギング サービスの一部は、TLS または Datagram TLS (DTLS) [14] を使用
- TCP 以外の転送手段を使用しているサービスの一部は、Google または Google 代行者が管理する物理境界の内側である場合、他の暗号化プロトコルまたはネットワークレベルの保護機能を使用

[13] 以前は他のプロトコルが使用されていましたが、現在は使用が中止されました。こうした古いプロトコルを使用しているジョブは、全体の 1% 未満です。
[14] Datagram TLS（DTLS）は、傍受や改ざんの防止策が施された通信を可能にすることで、データグラム ベースのアプリケーションのセキュリティを確保します。

VM と Google Cloud Platform サービス間の通信では、GFE との通信に、ALTS ではなく TLS が使用されます。これらの通信については、セクション 3.5 で説明します。

3.5 仮想マシンから Google Front End への通信の暗号化

VM から GFE へのトラフィックでは、Google のサービスに到達するために外部 IP を使用しますが、プライベート Google アクセス機能を構成すると、Google 専用の IP アドレスをリクエストに使用できます。

外部のユーザーから Google へのリクエストと同様に、VM から GFE までに関しても、デフォルトで TLS トラフィックがサポートされています。この接続は、その他の外部接続と同様の方法で確立されます。TLS の詳細については、セクション 3.1.1 をご覧ください。

4.転送されるデータの暗号化に関してユーザーが構成できるオプション

このドキュメントのセクション 3 では、転送されるデータを保護するために Google が利用しているデフォルトの保護機能について説明しました。このセクションでは、これらのデフォルトの保護機能についてユーザーが構成できる項目を説明します。

4.1 オンプレミスのデータセンターから Google Cloud への通信

4.1.1 GCLB 外部ロードバランサを使用する TLS

お客様のクラウド サービスで Google HTTPS ロードバランサまたは SSL プロキシ外部ロードバランサを使用している場合、GFE は、お客様がプロビジョニングし管理している SSL 証明書を使用して、ユーザーからの TLS 接続を終端します。 証明書のカスタマイズの詳細については、Google の SSL 証明書のドキュメントをご覧ください。

4.1.2 Google Cloud VPN を使用する IPSec トンネル

Google Cloud のお客様は、Google Cloud VPN を使用することで、お客様のオンプレミス ネットワークを Google Cloud Platform（GCP）Virtual Private Cloud（VPC）ネットワークへと IPSec VPN 接続（レイヤ 3）経由

で安全に接続できます。2 つのネットワーク間を移動するトラフィックは、一方の VPN ゲートウェイで暗号化され、もう一方の VPN ゲートウェイで復号化されます。これにより、インターネット上を移動するデータが保護されます。また、複数の VPN ゲートウェイを介して、複数の負荷分散されたトンネルを設定できます。Google Cloud VPN では、以下の方法でデータが保護されます。

- **お客様の VM から Cloud VPN への**パケットは、Google のネットワーク内に残ります。こうしたパケットは、Google または Google 代行者が管理する物理的境界の外側に出る場合、Google Cloud の仮想ネットワークによって暗号化されます。

- **Cloud VPN からお客様のオンプレミス VPN への**パケットは、IPSec トンネルを使用して暗号化され、認証されます。

- **お客様のオンプレミス VPN からオンプレミス ホストへの**パケットは、お客様のネットワークに導入されている任意の制御機能により保護されます。

VPN を設定するには、ホスト型サービスの VPN ネットワーク上に Cloud VPN ゲートウェイとトンネルを作成し、ネットワーク間のトラフィックを許可します。また、2 つの VPC 間に VPN を設定することもできます。

VPN トンネルに Internet Key Exchange[15] (IKE) バージョンを指定すると、ネットワークをさらにカスタマイズできます。IKE は IKEv1 と IKEv2 の 2 つのバージョンから選択可能で、それぞれが異なる暗号をサポートしています。IKEv1 を指定すると、AES-128-CBC を使用してパケットが暗号化され、SHA-1 HMAC[16] を介して整合性が確保されます。IKEv2 の場合は、さまざまな暗号の使用がサポートされています。いずれの場合も、Google Cloud VPN ではピア端末がサポートしている最も安全な共通プロトコルがネゴシエートされます。VPN の設定の詳細については、Google のドキュメント VPN の作成をご覧ください。

IPSec トンネルの代替手段としては、Google Cloud Dedicated Interconnect があります。Dedicated Interconnect は、お客様のオンプレミス ネットワークと Google のネットワーク間に、直接的な物理接続と RFC1918 通信を提供します。この接続を通じて転送されるデータは、デフォルトでは暗号化されないため、TLS を利用するなどの方法によって、アプリケーション レイヤで保護する必要があります。Google Cloud VPN と Google Cloud Interconnect は同じアタッチメント ポイントを使用していることか

Google Cloud のお客様は、Google Cloud VPN を使用することで、お客様のオンプレミス ネットワークを Google CloudPlatform (GCP) Virtual Private Cloud (VPC) ネットワークへと IPSec VPN 接続経由で安全に接続できます。

[15] Internet Key Exchange (IKE) は、IPSec プロトコル スイート内でセキュリティ アソシエーションを設定する際に使用するプロトコルです。
[16] HMAC-SHA-1 は、Google の研究者が発見した SHAttered 衝突のような SHA-1 衝突によって破壊されません。

ら、Dedicated Interconnect で IPSec VPN 暗号化を利用できますが、そのためにはサードパーティのソリューションを使用する必要があります。MACsec（レイヤ 2 保護）は、現時点ではサポートされていません。

4.2 ユーザーから Google Front End への通信

4.2.1 マネージド SSL 証明書: 無料の自動証明書

Google Cloud でアプリケーションを構築する場合は、使用する SSL 証明書を構成することで、GFE の TLS サポートを活用できます。たとえば、アプリケーションで TLS セッションを終端させることができます。この終端は、セクション 4.1.1 で説明している TLS 終端とは異なります。

Google では、Firebase Hosting と Google App Engine の両方のカスタム ドメインで、無料の自動 SSL 証明書も提供しています。これらの証明書は、Google がホストするプロパティにのみ使用できます。Google App Engine のカスタム ドメインを使用すると、独自の SSL 証明書を提供し、HTTPS Strict Transport Protocol（HSTS）ヘッダーを使用することもできます。

ドメインが Google のインフラストラクチャをポイントすると、Google からそのドメインの証明書がリクエストされて取得され、安全な通信が可能になります。Google は、TLS サーバーの秘密鍵（2,048 ビット RSA または secp256r1 ECC のいずれか）を管理し、お客様に代わって証明書を更新します。

Google Cloud でアプリケーションを構築する場合は、使用する SSL 証明書を構成することで、GFE の TLS サポートを活用できます。

4.2.2 Gmail での TLS 要件

セクション 3.1.1 で説明したように、Gmail ではデフォルトで TLS が使用されます。メールの最後のホップが TLS セッション上で行われたかどうかが記録され、表示されます[17]。Gmail ユーザーが別の Gmail ユーザーとメール交換した場合、それらのメールは TLS によって保護されるか、場合によってはアプリケーション内で直接送信されます。こうした場合、Gmail アプリケーションで使用されるリモート プロシージャ コールは、セクション 3.4 で説明したように ALTS で保護されます。Gmail では、他のメール プロバイダからの受信メールには TLS が適用されません。Gmail 管理者は、Gmail の構成により、すべての受信メールと送信メールについて安全な TLS 接続を要求できます。

[17] G Suite Enterprise の場合、これは UI には表示されません。ドメイン管理者は、メールログ検索を使用してドメインのデータを調べることができます。

4.2.3 Gmail S/MIME

Secure/Multipurpose Internet Mail Extensions（S/MIME）は、メールに認証、整合性、暗号化を提供するためのセキュリティ標準です。S/MIME 標準の実装では、メールを送信するユーザーに関連付けられる証明書は、公開 CA でホストすることが義務付けられています。

管理者は Gmail の構成により、送信メールの S/MIME を有効にし、コンテンツと添付ファイルのコンプライアンスに関するポリシーを設定し、受信メールと送信メールのルーティング ルールを作成できます。構成が完了した後、Gmail API を使用して、ユーザーの公開証明書を Gmail にアップロードする必要があります。Gmail 以外のユーザーの場合は、最初の S/MIME 署名付きメッセージを交換して S/MIME をデフォルトに設定する必要があります。

4.3 サービス間および VM 間の通信の暗号化

Istio は、サービスの検出と接続を簡略化するため、Google、IBM、Lyft などが開発したオープンソースのサービス メッシュです。Istio 認証では、サービス間で転送されるデータが自動的に暗号化され、関連する鍵と証明書が管理されます。Istio は、Google Container Engine および Google Compute Engine で使用できます。

ワークロードの相互認証と暗号化を実装する場合は、istio_auth を使用できます。具体的には、Kubernetes のワークロードの場合、Istio auth を使用すると、クラスタレベルの CA で証明書を生成して配布することができ、それらがポッド間の相互 Transport Layer Security（mTLS）に使用されます。

Istio は、サービスの検出と接続を簡略化するため、Google、IBM、Lyft などが開発したオープンソースのサービス メッシュです。Istio 認証では、サービス間で転送されるデータが自動的に暗号化され、関連する鍵と証明書が管理されます。

5.Google によるインターネット上の転送データの暗号化支援

セクション 3 およびセクション 4 では、転送されるお客様のデータに対して、Google Cloud で適用されるデフォルトの保護およびカスタマイズ可能な保護について説明しました。Google は、複数のオープンソース プロジェクトやその他の取り組みにより、転送されるデータの暗号化の利用およびインターネット全体でのデータ セキュリティも奨励しています。

5.1 Certificate Transparency

セクション 3.1 で説明したように、HTTPS を提供するには、信頼できるウェブ（パブリック）認証局（CA）による証明書をサイトがまず申請しなければなりません。認証局は、申請者がドメイン所有者の承認を受けているかどうかを検証するとともに、証明書に含まれるその他の情報がすべて正確であることを確認する責任があります。その後、この証明書がブラウザに表示され、ユーザーがアクセスしようとしているサイトが認証されます。HTTPS が適切に認証されるようにするため、CA は、ドメイン所有者が承認した証明書だけを発行することが重要です。

Certificate Transparency（CT、証明書の透明性）は、未承認の証明書や不正な証明書を CA が発行した場合に、サイト運営者とドメイン所有者がそれを検出できるようにするため、Google が 2013 年 3 月に開始した取り組みです。ドメイン所有者、CA、一般のユーザーが確認した信頼済みの証明書、CA の場合は自身が発行した証明書を、公開されている検証可能かつ追記専用の改ざん防止ログに記録できるという仕組みになっています。このログに記録された証明書は誰でも閲覧が可能で、情報が正確かつ承認済みであることを確認できます。

Certificate Transparency の最初のバージョンは、IETF の実験的 RFC である RFC 6962 で規定されました。Certificate Transparency の開発時に、Google は、証明書を記録できるオープンソースのログサーバーや Certificate Transparency ログを作成するためのツールなどをオープンソース化しました。さらに Google Chrome では、Extended Validation（EV）証明書や、過去に不正発行を行ったことがある CA から発行された証明書など、一部の証明書を一般公開することを義務付けています。2018 年以降、Chrome では新しいパブリック証明書はすべて開示が義務付けられます。

サイトの運営者は、Certificate Transparency を使用することで、自身のウェブサイトに対して不正な証明書が発行されたかどうかを検出できます。これを簡単に実施するための無償ツールは、Google の Certificate Transparency Report、Certificate Search、Facebook のツールなど、数多く存在しています。Certificate Transparency を使用していない場合でも、現在は多数のブラウザが Certificate Transparency を定期的に検査して、ユーザーがお客様のウェブサイトにアクセスする際に信頼している CA が業界の要件とベスト プラクティスを遵守していることを確認し、不正な証明書が発行されるリスクを低減しています。

5.2 HTTPS の利用の促進

セクション 3.1 で説明したとおり、Google のサイトやサービスでは、先進技術である HTTPS をデフォルトで提供するよう努めています。Google の目標は、すべてのプロダクトおよびサービスで 100% の暗号化を達成することです。このため、Google Cloud を含むすべてのプロパティの目標達成状況を追跡した、HTTPS 透明性レポートを毎年発行しています。さらに、HTTPS Strict Transport Protocol（HSTS）[18] をサポートしていないブラウザやその他のクライアント向けのソリューションなど、一部の Google サービスで暗号化のサポートを困難にしている技術的障壁の解消に取り組んでいます。google.com のホームページなど一部の Google サイトでは、ユーザーが HTTPS でのみサーバーに接続できるよう、HSTS を使用しています。

Google は、インターネット全般で HTTPS への移行が進んでいることを理解しており、この動きを促進するものとして以下の取り組みを進めています。

- デベロッパーに対し、HTTPS が重要な理由、HTTPS を有効にする方法、HTTPS を実装する際のベスト プラクティスについてのアドバイスを行っています。
- Chrome については、デベロッパーが自身のサイトの HTTP ステータスを評価できるよう、DevTools のセキュリティ パネルなどのツールを作成しました。
- 誰でも自分のウェブサイトの証明書を無償で入手できるLet's Encrypt イニシアチブを、資金面でサポートしています。Google の代表者は、Let's Encrypt の母体である Internet Security Research Group の技術諮問委員会に参加しています。

2016 年には、インターネット上の Google 以外の上位 100 サイトに関する「インターネットでの HTTPS 利用」の統計情報の公開を開始しました。Google は、この情報に基づき意識の向上を図るとともに、すべてのユーザーにとってインターネットをより安全な場所にすることを目指しています。2017 年 10 月、Chrome は Let's Encrypt に対する資金援助を正式に更新し、プラチナ スポンサーとなっています。

Google の目標は、すべてのプロダクトおよびサービスで 100% の暗号化を達成することです。このため、すべてのプロパティの目標達成状況を追跡した、HTTPS 透明性レポートを毎年発行しています。

[18] HTTPS Strict Transport Protocol は、ウェブサイトが自身へのアクセスをセキュアな接続を介してのみ可能であると宣言したり、ユーザーがユーザー エージェントに対し、セキュアな接続を介してのみ特定のサイトと通信するように指示したりすることを可能にするメカニズムです。

5.3 セキュアな SMTP の利用促進: Gmail の指標

ほとんどのメールは、デフォルトで暗号化を使用せずにメールを送信する SMTP (Simple Mail Transfer Protocol) を利用して交換されています。メールを暗号化するには、メール プロバイダが TLS などのセキュリティ管理を実装している必要があります。

セクション 3.1 で説明したとおり、Gmail ではデフォルトで TLS が使用されます。また、セクション 4.2.2 では、送受信されるメールに関して、Gmail 管理者がどのように TLS 保護の利用を適用できるのかを説明しました。Google では、HTTPS の透明性に関する取り組みと同様に、Gmail でも受信メールへの TLS 利用に関するデータを提供しています。このデータは、Google の Safer Email Transparency Report に掲載されています。

Google は、IETF をはじめとする業界の主要団体と協力しながら、SMTP STS の開発を主導しています。SMTP STS は HTTPS の HSTS に似たもので、暗号化されたチャネルでのみ、SMTP の使用を強制します。

5.4 Chrome API

2015 年 2 月、Chrome は、安全な送信元に対してのみ、いくつかの優れた新機能が使用可能になると発表しました[19]。こうした新機能には、個人情報の処理や、ユーザー端末上のセンサーへのアクセスなどがあります。安全でない送信元に対しては、Chrome 50 の位置情報を皮切りに、こうした機能のサポートを終了し始めています。

6.転送データの暗号化の継続的な革新

6.1 Chrome のセキュリティに関するユーザー エクスペリエンス

Google Chrome は、サイトへの接続が安全であるかどうかをユーザーがすぐに理解できる、UI を活用したセキュリティ情報の表示に関して、業界を先導する存在となっています。ユーザーは、この情報を利用して、データをいつどのように共有するかを決定できます。Chrome は広範にわたるユーザー調査を実施しており、その結果はピアレビュー済みの記事で公開されています。

Google Chrome は、サイトへの接続が安全であるかどうかをユーザーがすぐに理解できる、UI を活用したセキュリティ情報の表示に関して、業界を先導する存在となっています。

[19] 安全な送信元とは、特定のスキーム、ホスト、ポートのパターンに一致する接続のことです。

ユーザー保護の強化を促進するため、Chrome は、2017 年末までにすべての HTTP 接続を「安全ではない」とマークすることを発表しました。Chrome 56 以降では、HTTP ページにパスワードまたはクレジット カードのフィールドを含むフォームがある場合、デフォルトで警告が表示されるようになります。Chrome 62 では、ユーザーが HTTP ページにデータを入力したときと、シークレット モードでアクセスしたときに、すべての HTTP ページで警告が表示されるようになります。最終的に、HTTP で配信されるすべてのページで警告が表示されるようになる予定です。

Chrome で特定の構成がユーザーに対してどのように表示されるのかを確認するには、BadSSL ツールを使用してください。

6.2 Key Transparency

メッセージの暗号化が広く採用されることを阻む大きな要因の 1 つは、公開鍵交換の難しさにあります。つまり、通信の相手となる新しいユーザーの公開鍵を確実に見つけるにはどうすればよいかということです。この問題を解決するため、2017 年 1 月、Google は Key Transparency を発表しました。これは、公開鍵を配布するための一般的かつ安全で監査可能な手段を提供する、オープンなフレームワークです。このフレームワークを導入すると、ユーザーが手動で鍵検証を行う必要がなくなります。Key Transparency は、E2E や OpenPGP のメール暗号化など、通信におけるユーザーの公開鍵の配布を主な目的としています。Key Transparency の設計は、鍵の復旧と配布に関する新しいアプローチであり、Certificate Transparency と CONIKS から得られた分析情報に基づいています。

Key Transparency の開発はオープンソースであり、大規模な Merkle ツリーを使用して実装されています。Key Transparency Verification を導入すると、アカウントの所有者は、アカウントに関連付けられている鍵およびアカウントが有効で安定している期間を確認できるようになります。Google による Key Transparency の取り組みの長期的目標は、誰でも Key Transparency サーバーを実行し、任意の数のアプリケーションに簡単に統合できるようにすることです。

6.3 ポスト量子暗号

Google が目指しているのは、転送されるデータの暗号化に関して、業界のリーダーであり続けることです。この目的を果たすため、Google は、ポスト量子暗号の分野での取り組みに着手しています。このタイプの暗号を利用すると、巧妙な量子攻撃に弱い既存の基礎暗号を、より堅牢とされるポスト量子暗号に置き換えることが可能になります。2016年

7 月、Google は、Chrome のデベロッパー版で New Hope ポスト量子暗号アルゴリズムを使用して、このようなアルゴリズムの導入可能性についての実験を行ったことを発表しました。この他にも、Google の研究者は、他の実用的なポスト量子鍵交換プロトコルに関する論文を発表しています。

付録

Google Cloud のセキュリティとコンプライアンスに関する一般的な情報については、Google Cloud Platform のウェブサイトおよび G Suite のウェブサイトで、Google インフラストラクチャ セキュリティ設計の概要や公開 SOC3 監査レポートなど、セキュリティに関するセクションをご覧ください。

Google Cloud での Application Layer Transport Security

Google Cloud ホワイトペーパー

Cesar Ghali、Adam Stubblefield、
Ed Knapp、Jiangtao Li、
Benedikt Schmidt、Julien Boeuf

Google Cloud

Google Cloud

目次

CIO レベルの概要

- Google の Application Layer Transport Security（ALTS）は、Google が開発した相互認証および転送暗号化システムで、通常は Google のインフラストラクチャ内におけるリモート プロシージャ コール（RPC）通信を保護するために使用されます。ALTS は概念的には相互認証 TLS と似ていますが、Google のデータセンター環境のニーズを満たすように設計され最適化されています。

- ALTS の信頼モデルは、クラウドのようなコンテナ化されたアプリケーションに合わせて調整されています。ID は特定のサーバー名またはホストではなく、エンティティにバインドされます。この信頼モデルは、ホスト間でのシームレスなマイクロサービス複製、負荷分散、および再スケジューリングを容易にします。

- ALTS は handshake プロトコル（セッション再開あり）とレコード プロトコルという 2 つのプロトコルに依存しています。これらのプロトコルはセッションを確立し、認証し、暗号化し、再開する方法を管理します。

- ALTS は Google が使用しているカスタムのトランスポート層セキュリティ ソリューションです。Google では本番環境に合わせて ALTS を調整しているため、ALTS と業界標準である TLS との間にはいくらかのトレードオフがあります。詳細については、トレードオフのセクションを参照してください。

1.はじめに

Google の本番環境システムは、[1] 秒あたり 0 (1010) 個のリモート プロシージャ コール (RPC) をまとめて発行するマイクロサービス1 のコンスタレーションで構成されています。Google のエンジニアが本番環境ワークロード[2] をスケジュールすると、そのワークロードで発行または受信された RPC は、デフォルトで ALTS によって保護されます。この構成不要の自動的な保護は、Google の Application Layer Transport Security (ALTS) によって提供されます。ALTS では RPC にこの自動保護が与えられることに加えて、本番環境マシン間での簡単なサービス複製、負荷分散、再スケジューリングが容易になります。このホワイトペーパーでは ALTS について説明し、Google の本番環境インフラストラクチャ上におけるそのデプロイについて見ていきます。

対象者: このドキュメントは、Google で認証と転送のセキュリティがどのように実行されているかについて関心をお持ちの、インフラストラクチャ セキュリティの専門家を対象としています。

前提条件: 読者はこの概要の知識に加えて、Google でのクラスタ管理についても基本的な理解を有しているものと前提します。

Google のエンジニアが本番環境ワークロードをスケジュールすると、そのワークロードで発行された RPC に Google の Application Layer Transport Security (ALTC) による構成不要の自動的な保護が提供されます。

2.アプリケーション レベルのセキュリティと ALTS

ウェブブラウザから VPN に至るまで、多くのアプリケーションは TLS (Transport Layer Security) や IPSec などの安全な通信プロトコルに依存して、転送中のデータを保護しています[3]。Google では、アプリケーション レイヤで動作する相互認証および転送暗号化システムである ALTS を使用して、RPC 通信を保護しています。アプリケーション レベルのセキュリティを使用すると、各アプリケーションが認証されたリモートピア ID を持つことができ、それを利用してきめ細かな認証ポリシーを実装することが可能になります。

2.1 TLS を使用しない理由

現在のインターネットトラフィックの大部分がTLSを使用して暗号化されている中で、GoogleがALTSのようなカスタムのセキュリティソリューションを使用していることは奇妙に思われるかもしれません。ALTSは2007年、Google が開発を開始しました。当時 TLS は、Google の最低限のセキュリティ基準を満たさない数多くのレガシー プロトコルのサポートにバンドルされていました。Google では、必要な TLS コンポーネントを採用し望ましいものを実装することで独自のセキュリティ ソリューションを設計するこ

[1] **マイクロサービス**は、ビジネス機能を実装する疎結合サービスの集合体としてアプリケーションを構築するアーキテクチャ スタイルです。
[2] **本番環境ワークロード**は、Google のエンジニアが Google のデータセンターで実行をスケジューリングするアプリケーションです。
[3] Google で転送中のデータがどのように保護されるかの詳細については、ホワイトペーパー Google Cloud での転送データの暗号化をご覧ください。

とも可能でしたが、よりGoogleに適したシステムを一から構築するほうが、既存のシステムにパッチを当てるよりもメリットが大きいと判断しました。さらに ALTS は Google のニーズにより適しており、過去の状況を鑑みても、古い TLS よりも安全性に優れています。以下に、TLS と ALTS の主な違いを示します。

- HTTPS のセマンティクスを使用した TLS の信頼モデル[4] と ALTS には大きな違いがあります。前者では、サーバー ID は特定の名前とそれに対応する命名スキームにバインドされています。ALTS では、同じ ID を複数の命名スキームで使用できます。このレベルの間接性によって柔軟性が向上し、ホスト間でのマイクロサービス複製、負荷分散、再スケジューリングのプロセスが大幅に簡略化されます。

- TLS と比較して、ALTS は設計と実装が容易です。その結果、ソースコードの手動検査や広範なファジングによってバグやセキュリティの脆弱性を監視することがより簡単になっています。

- ALTS ではプロトコル バッファを使用して証明書やプロトコル メッセージがシリアル化されますが、TLS では ASN.1 でエンコードされた X.509 証明書が使用されます。Google の本番環境サービスの大部分は、プロトコル バッファを使用して通信（および場合によってはストレージ）を行っているため、ALTS のほうが Google の環境により適したものとなっています。

2.2 ALTS の設計

ALTS は高い信頼性を誇るシステムとして設計されており、ユーザーの関与を最小限に抑えたサービス間認証とセキュリティを実現しています。これを達成するため、以下のような特性が ALTS の設計の一部となっています。

- **透明性:** ALTS 構成はアプリケーション レイヤに対して透過的です。サービス RPC はデフォルトで、ALTS によって保護されます。そのためアプリケーション デベロッパーは、認証情報の管理やセキュリティ構成について心配することなく、サービスの機能ロジックに集中できます。サービス間接続の確立時、ALTS は認証されたリモートピア ID 情報をアプリケーションに提供します。これを使用することで、きめ細かな認証チェックと監査が可能になります。

- **最先端の暗号化:** ALTS で使用されているすべての暗号プリミティブおよびプロトコルは、現在の既知の攻撃について最新のものです。ALTS は Google が管理するマシン上で動作します。つまり、サポートされているすべての暗号プロトコルは、簡単にアップグレードして迅速に導入することができます。

ALTS は高い信頼性を誇るシステムとして設計されており、ユーザーの関与を最小限に抑えたサービス間認証とセキュリティを実現しています。

[4] 信頼モデルとは、セキュリティ プロトコルが認証情報と ID を識別し、分配し、回転させるメカニズムです。

- **ID モデル:** ALTS は認証を、ホスト名ではなく主に ID で実行します。Google ではすべてのネットワーク エンティティ（企業ユーザー、物理マシン、本番環境のサービスやワークロードなど）に、関連付けられた ID があります。サービス間のすべての通信は相互に認証されます。

- **キーの配布:** ALTS は ID を持つ各ワークロードに依存します。ID は一連の認証情報として表されます。こうした認証情報は、ユーザーの関与なしに、初期化時に各ワークロード内でデプロイされます。それと並行して、それらの認証情報の信頼ルートと信頼チェーンがマシンとワークロードに対して確立されます。このシステムでは、アプリケーション開発者の介入なしに、自動での証明書のローテーションと取り消しが可能になっています。

- **スケーラビリティ:** ALTS は Google の大規模なインフラストラクチャをサポートするため、非常にスケーラブルに設計されています。この要件により、効率的なセッション再開が開発されました。

- **長時間接続:** 認証された鍵交換の暗号化オペレーションは、コンピューティング コストが高くなります。Google のインフラストラクチャの規模に対応するため、初回の ALTS handshake の後は、接続を長期間維持することで、システムの全体的なパフォーマンスを向上させることができます。

- **シンプルさ:** TLS はデフォルトで、レガシーのプロトコル バージョンと下位互換性をサポートしています。一方 ALTS は、Google がクライアントとサーバーの両方を制御しており、それらがネイティブで ALTS をサポートするように設計されているため、はるかにシンプルになっています。

3. ALTS の信頼モデル

ALTS は認証を、ホストではなく主に ID で実行します。Google では、すべてのネットワーク エンティティ（企業ユーザー、物理マシン、本番環境サービスなど）に、関連付けられた ID があります。これらの ID は ALTS 証明書に埋め込まれており、セキュアな接続の確立時のピア認証に使用されます。Google が追求しているモデルは、Google の本番環境サービスを、サイト信頼性エンジニア（SRE）[5]が管理できる本番環境エンティティとして実行することです。これらの本番環境サービスの開発バージョンは、SRE と開発者の両者が管理できるテスト エンティティとして実行されます。

ALTS は認証を、ホストではなく主に ID で実行します。Google では、すべてのネットワーク エンティティ（企業ユーザー、物理マシン、本番環境サービスなど）に、関連付けられた ID があります。

[5] 一部のサービスはデベロッパーが直接管理しています。

たとえば、「`service-frontend`」と「`service-backend`」という 2 つのサービスを含むサービスがあるとします。SRE　はこれらのサービスの本番環境バージョンである、「`service-frontend-prod`」および「`service-backend-prod`」を起動できます。開発者はテストのため、これらのサービスの開発バージョン「`service-frontend-dev`」と「`service-backend-dev`」をビルドして起動することができます。本番環境サービスの承認構成は、各サービスの開発バージョンを信頼しないように構成されます。

3.1 ALTS の認証情報

ALTS の認証情報には 3 種類あり、すべてプロトコル バッファ メッセージ形式で表現されます。

- **マスター証明書:** リモートの署名サービスによって署名され、handshake 証明書の検証に使用されます。マスター証明書には、RSA 鍵ペアなどのマスター秘密鍵に関連付けられた公開鍵が含まれています。この秘密鍵は handshake 証明書への署名に使用されます。これらの証明書は、以下で説明する ALTS ポリシーと組み合わせて使用する場合、基本的に制限付きの中間認証局（CA）の証明書となります。マスター証明書は通常、Borgmaster[6] のようなコンテナ化されたワークロードの本番環境マシンとスケジューラに対して発行されます。

- **handshake 証明書:** マスター秘密鍵によってローカルに作成および署名されます。この証明書には、静的な Diffie-Hellman（DH）パラメータや handshake 暗号など、ALTS handshake（セキュアな接続の確立）の際に使用されるパラメータが含まれています。また handshake 証明書には、その派生元となるマスター証明書、すなわち handshake 証明書に署名するマスター秘密鍵に関連付けられた証明書が含まれています。

- **再開鍵:** 再開チケットを暗号化するために使用される秘密鍵です。この鍵は、同じデータセンター セル内で同じ ID で実行されるすべての本番環境ワークロードに対して一意でありそれらによって共有される、再開識別子 ID_R で識別されます。ALTS でのセッション再開の詳細については、セッションの再開をご覧ください。

図 1 は、署名サービスの検証鍵、マスター証明書、handshake 証明書で構成される、ALTS 証明書チェーンを示しています。署名サービスの検証鍵は ALTS における信頼のルートであり、Google の本番環境ネットワークや企業ネットワーク内のすべての Google マシンにインストールされます。

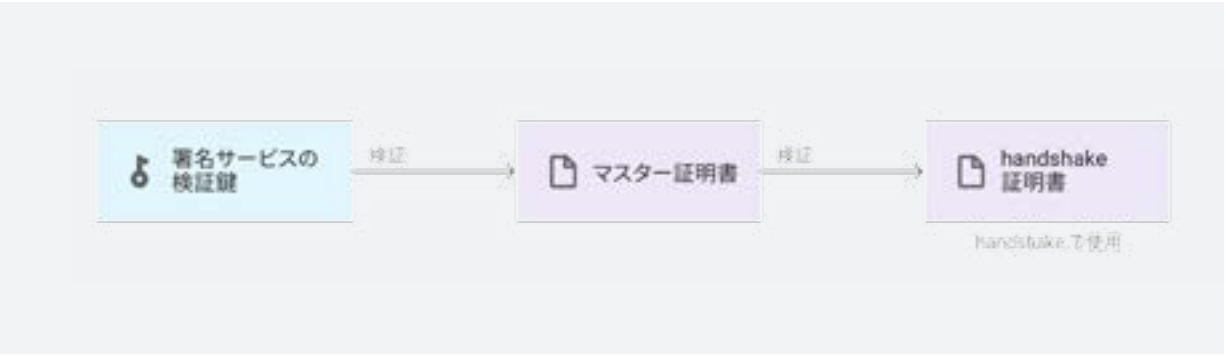


図 1:

ALTS
証明書チェーン

ALTS では署名サービスがマスター証明書を認証し、次にそのマスター証明書が handshake 証明書を認証します。handshake 証明書はマスター証明書より頻繁に作成されるため、このアーキテクチャによって署名サービスの負荷が軽減されます。Google では特に handshake 証明書について、証明書のローテーションが頻繁に発生します[7]。この頻繁なローテーションにより、handshake 証明書に含まれる静的な鍵交換ペアが補われます[8]。

3.1.1 証明書の発行

ネットワーク上のエンティティを ALTS のセキュアな handshake に参加させるには、エンティティに handshake 証明書をプロビジョニングする必要があります。発行元はまず、署名サービスによって署名されたマスター証明書を取得し、それを必要に応じてエンティティに渡します。次に handshake 証明書が作成され、関連付けられたマスター秘密鍵によって署名されます

発行元は通常、マシンや人に対して証明書を発行するときはGoogle 内部の認証局（CA）、ワークロードに対して証明書を発行するときは Borgmaster になります。ただしこれは、たとえばテスト データセンターセルに対しては制限付き Borgmaster など、他のエンティティであってもかまいません。

[6] Borgmaster は、Google の本番環境ワークロードのスケジューリングと初期化を担当します。詳細については、Borg を使用した Google の大規模クラスタ管理をご覧ください。
[7] 証明書のローテーション頻度の詳細については、「Google Cloud での転送データの暗号化」をご覧ください。
[8] 鍵が侵害された場合、攻撃者が確認できるのは、この鍵ペアの存続期間中のトラフィックだけです。

図 2 は、署名サービスを使用してマスター証明書を作成する方法を示しています。このプロセスは以下の手順で構成されます。

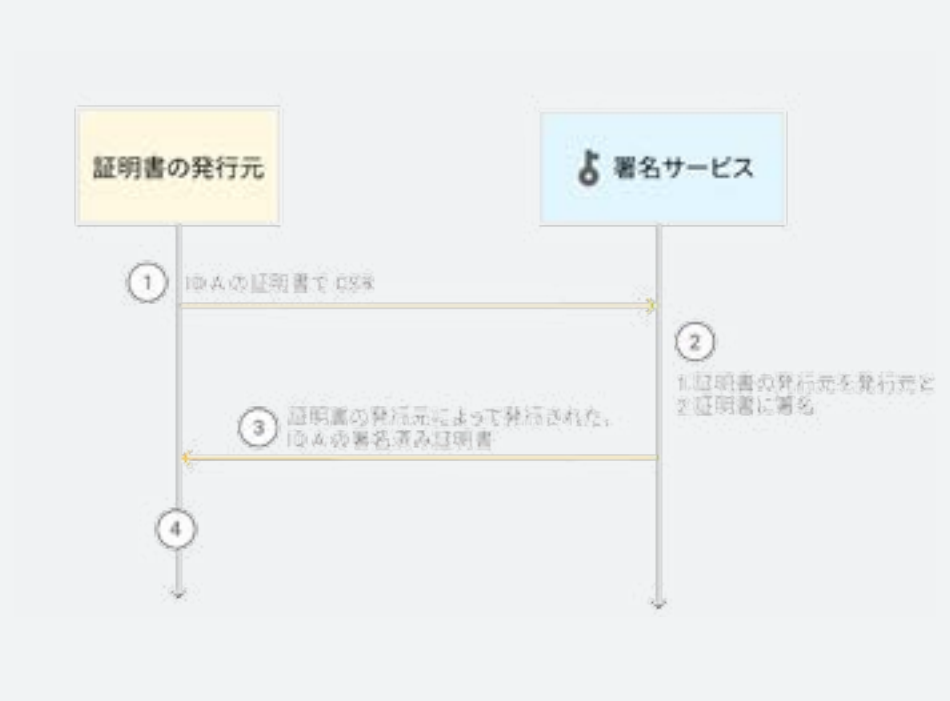


図 2:

証明書の発行

1. 証明書の発行元が、証明書署名リクエスト（CSR）を署名サービスに送信します。このリクエストは ID A の証明書を作成するよう署名サービスに要求するものです。この ID はたとえば企業ユーザーや、Google の本番環境サービスの ID などにできます。

2. 署名サービスは証明書の発行元（CSR に含まれる）を、リクエスト元（この場合は証明書の発行元）に設定して署名します。対応する署名サービスの公開（検証）鍵が、すべての Google マシンにインストールされていることを思い出してください。

3. 署名サービスが、署名された証明書を返送します。

4. ID A の handshake 証明書が作成され、マスター証明書に関連付けられた秘密鍵によって署名されます。

上記のプロセスに示されるように、ALTS　では証明書の発行元と署名者は 2 つの異なる論理エンティティになります。この場合、発行元は証明書の発行元エンティティであり、署名者は署名サービスです。

ALTS には、人、マシン、ワークロードという 3 つの一般的な証明書カテゴリがあります。以下のセクションでは、これらの証明書がそれぞれ ALTS でどのように作成され使用されるかを概説します。

3.1.2 人の証明書

Google では ALTS を使用して、人間のユーザーが本番環境サービスに対して発行する RPC を保護しています。RPC を発行するには、ユーザーが有効な handshake 証明書を提供する必要があります。たとえば A さんがアプリケーションを使用して、ALTS で保護される RPC を発行する場合、Google の内部 CA に対して認証することができます。この場合はユーザー名、パスワード、および二要素認証を使用して CA への認証を行います。このオペレーションの結果、A さんは 20 時間有効な handshake 証明書を取得できます。

3.1.3 マシン証明書

Google のデータセンターのすべての本番環境マシンには、マシンのマスター証明書があります。この証明書はマシン管理デーモンなど、そのマシン上のコア アプリケーションの handshake 証明書を作成するために使用されます。マシン証明書に埋め込まれている主な ID は、マシンの一般的な目的を表します。たとえば、異なる種類の本番環境ワークロードおよび開発ワークロードを実行するために使用されるマシンは、異なる ID を持つことができます。マスター証明書は検証済みのソフトウェア スタックを実行しているマシンでのみ使用できます。場合によっては、この信頼のルートがカスタムのセキュリティ ハードウェアにあることもあります[9]。本番環境マシンのマスター証明書はすべて CA によって発行され、数か月ごとにローテーションされます。また、すべての handshake 証明書は数時間ごとにローテーションされます。

3.1.4 ワークロード証明書

ALTS の重要な利点は、ワークロード ID の発想で機能しているため、マシン間での簡単なサービス複製、負荷分散、再スケジューリングが容易になることです。Google の本番環境ネットワークでは、Borg[10] というシステムを使用してクラスタ管理とマシンリソースの割り当てを大規模に行っています。Borg による証明書の発行方法は、ALTS のマシンに依存しないワークロード ID の実装の一部です。

Google の本番環境ネットワークの各ワークロードは、Borg セルで実行されます。各セルには Borgmaster という論理的に一元化されたコントローラと、そのセル内の各マシン上で実行される Borglets という複数の

[9] Titan の詳細: プレーン テキストのセキュリティ。
[10] Borg を使用した Google の大規模クラスタ管理

エージェント プロセスが含まれています。ワークロードは Borgmaster が発行した、関連付けられたワークロード handshake 証明書によって初期化されます。図 3 は、Borg を使用した ALTS でのワークロード認証のプロセスを示しています。

1. 各 Borgmaster には、マシンマスター証明書と、関連付けられた秘密鍵（図には示していません）があらかじめインストールされています。

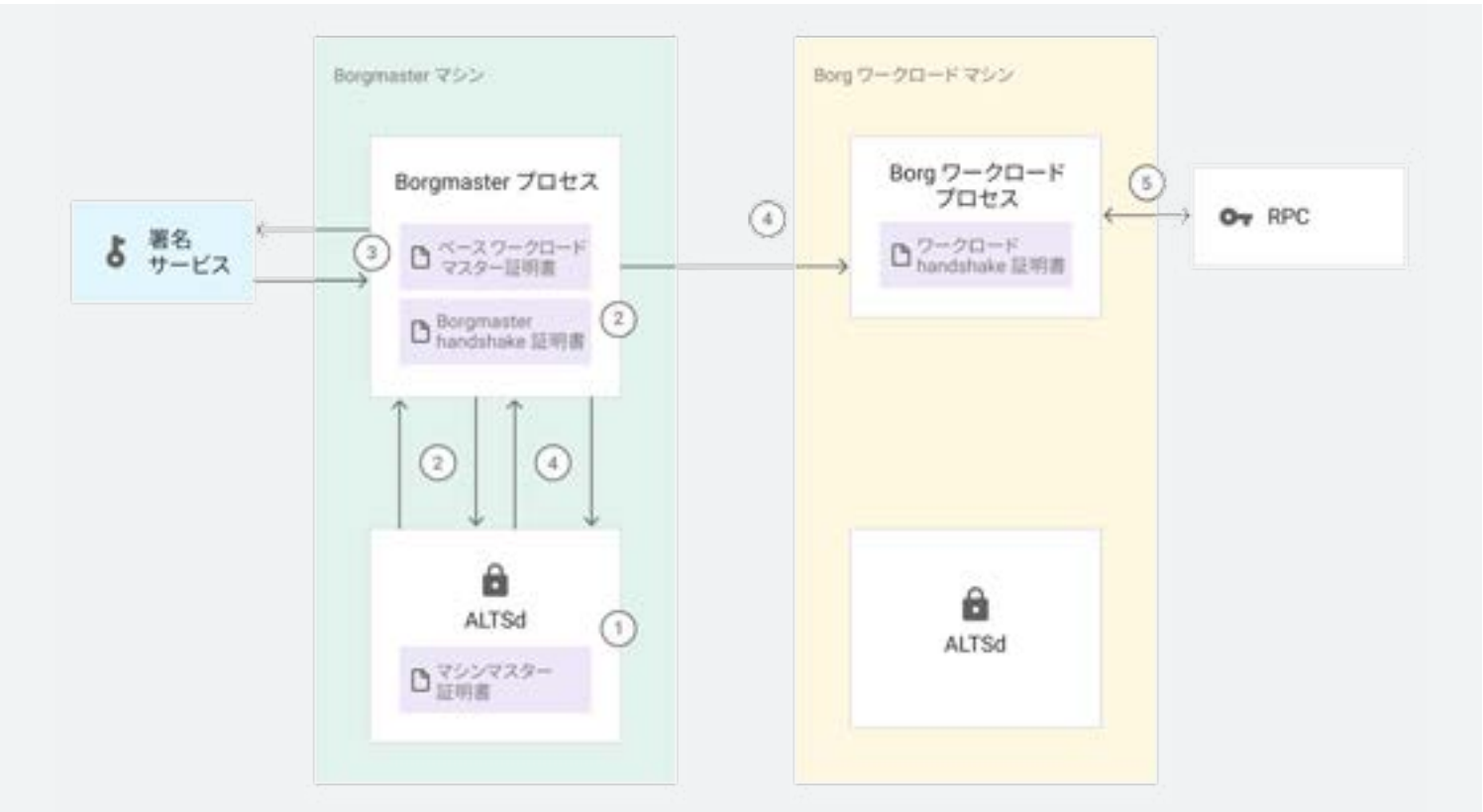
2. ALTSd[11] は Borgmaster の handshake 証明書を生成し、マシンマスター秘密鍵を使用してそれに署名します。この handshake 証明書により、Borgmaster は ALTS で保護される RPC を発行できます。

3. Borgmaster がベース ワークロード マスター証明書と、対応する秘密鍵を作成します。Borgmaster は、署名サービスによって署名されたこのベース ワークロード マスター証明書を取得するためのリクエスト（ALTS で保護される RPC）を開始します。その結果、署名サービスは Borgmaster をこの証明書に発行元として記録します。

Borgmaster はこれで、ALTS を使用する必要のあるワークロードをスケジューリングできます。以下の手順は、クライアントが Borg 上で特定の ID として実行するワークロードをスケジューリングするときに発生します。

図 3:

Google の本番環境ネットワークにおける handshake 証明書の作成



[11] ALTSd: さまざまな ALTS 機能の中でも、handshake 証明書の作成を担当するデーモン。

4. Borgmaster はクライアントが、ワークロードをワークロード構成で指定された ID として実行する権限を持っていることを確認します。問題がなければ、Borgmaster は Borglet に Borg ワークロードをスケジューリングし、ワークロード handshake 証明書とそれに対応する秘密鍵を発行します。この証明書はベース ワークロード マスター証明書からチェーンされています。その後、ワークロード handshake 証明書とその秘密鍵は、Borglet に（Borgmaster と Borglet の間の相互認証された ALTS 保護チャネルを介して）安全に送信されます。Borgmaster はベース ワークロード マスター証明書をローテーションし、実行中のすべてのワークロードについて、handshake 証明書を約 2 日ごとに再発行します。さらに同じセル内で同じユーザーとして実行されている各ワークロードは、Borgmaster によってプロビジョニングされた同じ再開鍵と識別子（ID_R）を受け取ります。

5. ワークロードが ALTS で保護される RPC を作成する必要がある場合は、handshake プロトコルでワークロード handshake 証明書が使用されます。ID_R はセッション再開を開始するための handshake の一部としても使用されます。ALTS でのセッション再開の詳細については、セッションの再開をご覧ください。

3.2 ALTS ポリシーの適用

ALTS ポリシーは、どの発行元がどの ID について特定のカテゴリの証明書を発行する権限を持っているかをリストしたドキュメントです。これは Google の本番環境ネットワーク上のすべてのマシンに配布されます。たとえば ALTS ポリシーでは、CA はマシンおよび人に対して証明書を発行できます。また Borgmaster は、ワークロードに対して証明書を発行できます。

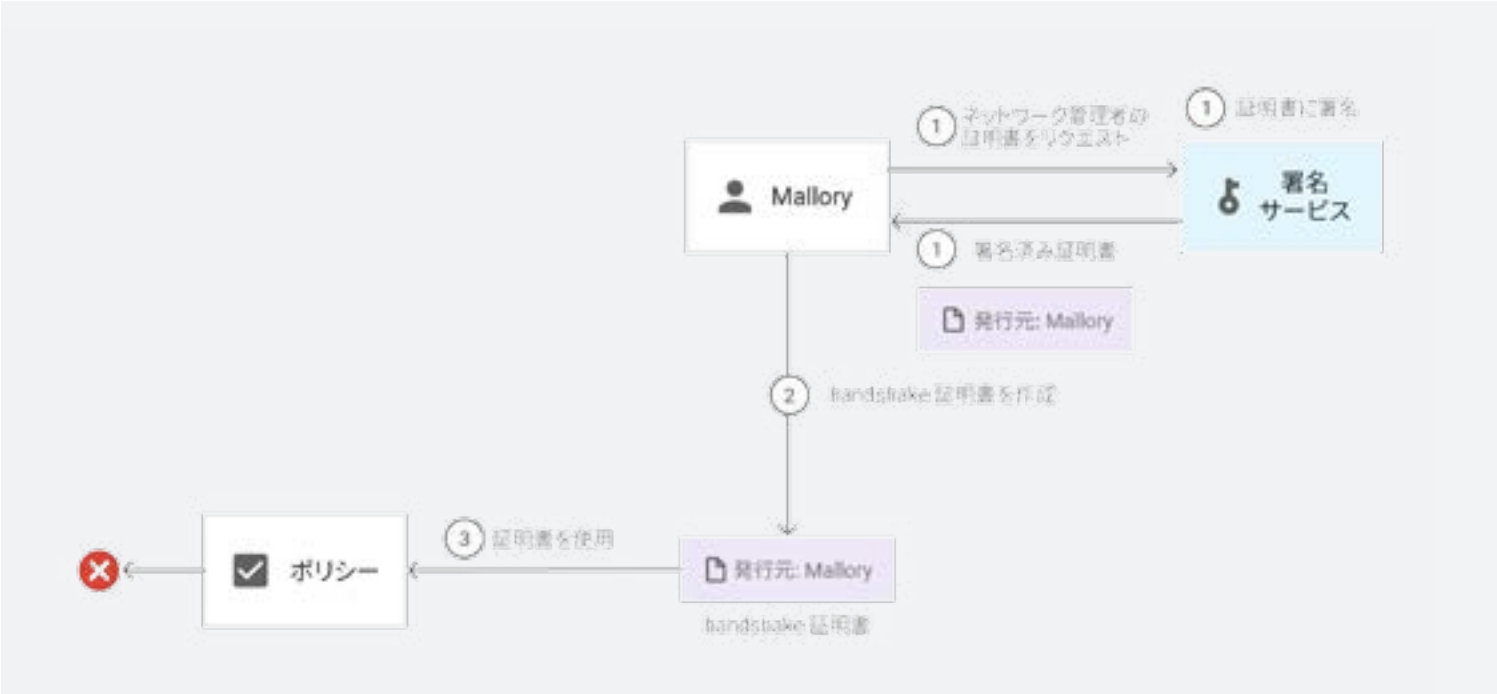
証明書の発行時とは対照的に、証明書の検証時のポリシー適用は、さまざまな種類のデプロイに対して異なるポリシーを適用できるため、より柔軟なアプローチであることがわかっています。たとえばテストクラスタのポリシーは、本番環境クラスタのポリシーよりも制限の緩いものにすることが可能です。

ALTS handshake の際、証明書の検証には ALTS ポリシーのチェックが含まれます。このポリシーでは、検証された証明書にリストされている発行元が、その証明書を発行する権限を持つことが保証されます。そうでない場合、証明書は拒否され、handshake 処理は失敗します。図 4 は ALTS

でポリシー適用がどのように機能するかを示しています。図 2 のシナリオに従って、ここでは　Mallory（特権をエスカレーションしようと考えている企業ユーザー）が、ネットワーク管理者に対し、マスター証明書を発行しようとしているとします。これはネットワークを再構成できる強力な ID です。この場合、Mallory が ALTS ポリシーでこのオペレーションの実行を許可されないことは言うまでもありません。

図 4:

証明書の発行と利用



1. 佐藤さんがネットワーク管理者 ID 用のマスター証明書を発行し、そこに署名サービスによる署名を受けます。これは図 2 の最初の 3 つのステップと同様です。

2.　佐藤さんは作成したマスター証明書に関連付けられたマスター秘密鍵を使用して、ネットワーク管理者の handshake 証明書をローカルに作成し署名します。

3.　作成した　handshake　証明書を使用して佐藤さんがネットワーク管理者の ID を偽装しようとした場合は、佐藤さんが通信しようとしている相手側の　ALTS　ポリシー施行者が、そのオペレーションをブロックします。

3.3 証明書の失効

Google では、証明書は期限切れになったか、Google の証明書失効リスト（CRL）に含まれている場合、無効とされます。このセクションでは、Google　　内部の証明書失効メカニズムの設計について説明します。このメカニズムは、本ドキュメントの作成時点においてはまだデプロイテスト中です。

人間の企業ユーザーに対して発行されるすべての証明書には、日次の有効期限タイムスタンプが付いているため、ユーザーは毎日再認証を行う必要があります。本番環境マシンに対して発行される証明書の多くは、有効期限タイムスタンプを使用していません。Google　では本番環境用証明書の期限満了について、タイムスタンプに依存することを避けています。これはクロック同期の問題による機能停止につながるおそれがあるからです。代わりに Google では、証明書のローテーションおよびインシデント対応処理の信頼できる情報源として、CRL を使用しています。図 5 は CRL がどのように機能するかを示したものです。

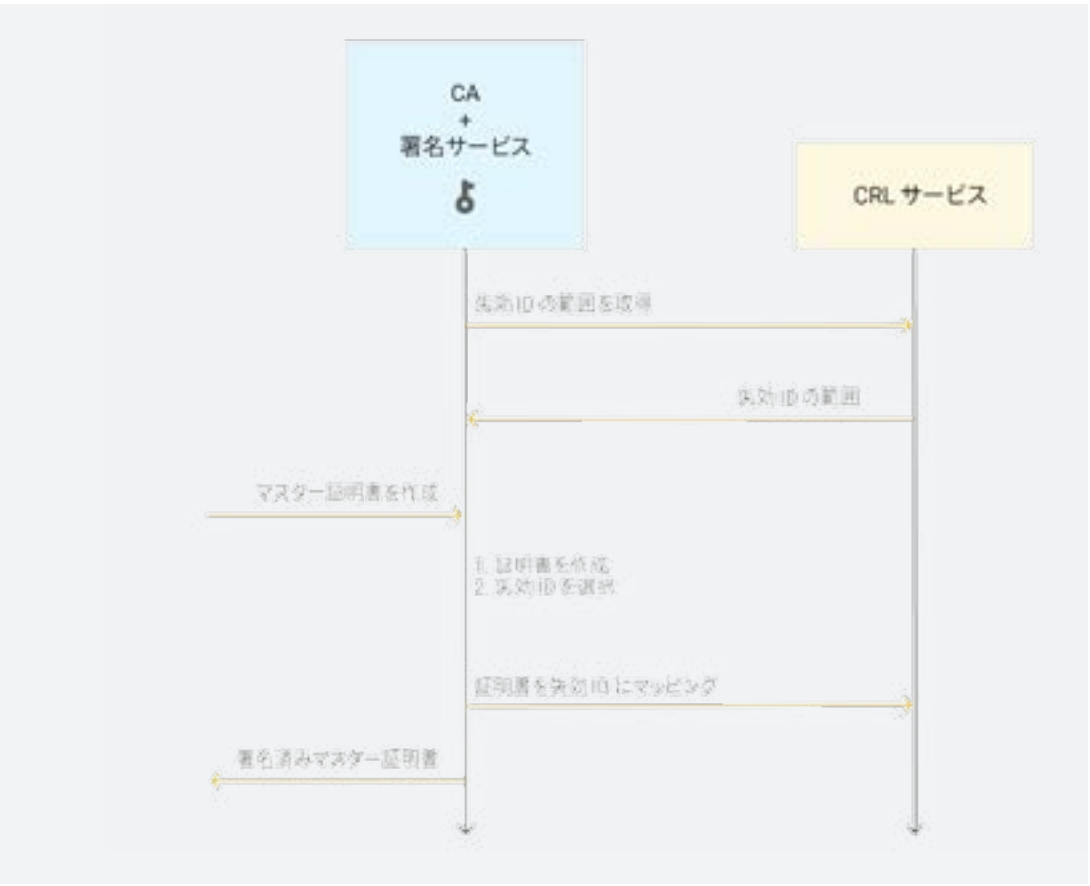


図 5:

失効 ID を使用したマスター証明書の作成

[12] 実際には、CA は署名サービスの秘密鍵へのアクセス権を持っているため、2 つの論理エンティティが単一の物理エンティティとなります。

1. CA のインスタンスが初期化されると12、CA は CRL サービスに連絡し、失効 ID の範囲を要求します。失効 ID は、8 ビットの証明書カテゴリ（人の証明書やマシンの証明書など）と 56 ビットの証明書 ID という 2 つの要素を含む、長さ 64 ビットの ID です。CRL サービスはこれらの ID の範囲を選択し、それを CA に返します。

2. CA はマスター証明書のリクエストを受信すると、証明書を作成し、そこに範囲から選択した失効 ID を埋め込みます。

3. それと同時に、CA は新しい証明書を失効 ID にマッピングし、その情報を CRL サービスに送信します。

4. CA がマスター証明書を発行します。

handshake 証明書に割り当てられる失効 ID は、証明書の使用方法によって異なります。たとえば人間の企業ユーザーに対して発行される handshake 証明書は、ユーザーのマスター証明書の失効 ID を継承します。Borg ワークロードに対して発行される handshake 証明書の場合、失効 ID は Borgmaster の失効 ID の範囲によって割り当てられます。この ID 範囲は、図 5 と同様のプロセスで CRL サービスによって Borgmaster に割り当てられます。ピアが ALTS handshake に関与している場合は、ピアが CRL ファイルのローカルコピーをチェックして、リモートピア証明書が失効していないことを確認します。

CRL サービスはすべての失効 ID を、ALTS を使用するすべての Google マシンに対して push 可能な単一のファイルにコンパイルします。CRL データベースは数百メガバイトですが、生成される CRL ファイルはさまざまな圧縮技術が利用されるためわずか数メガバイトになります。

4. ALTS プロトコル

ALTS は、handshake プロトコル（セッション再開あり）とレコード プロトコルという 2 つのプロトコルに依存しています。このセクションでは、各プロトコルの概要を詳しく説明します。ただしこれらの概要は、プロトコルの詳細な仕様と解釈されるべきものではありません。

ALTS は、handshake プロトコル（セッション再開あり）とレコード プロトコルという 2 つのプロトコルに依存しています。

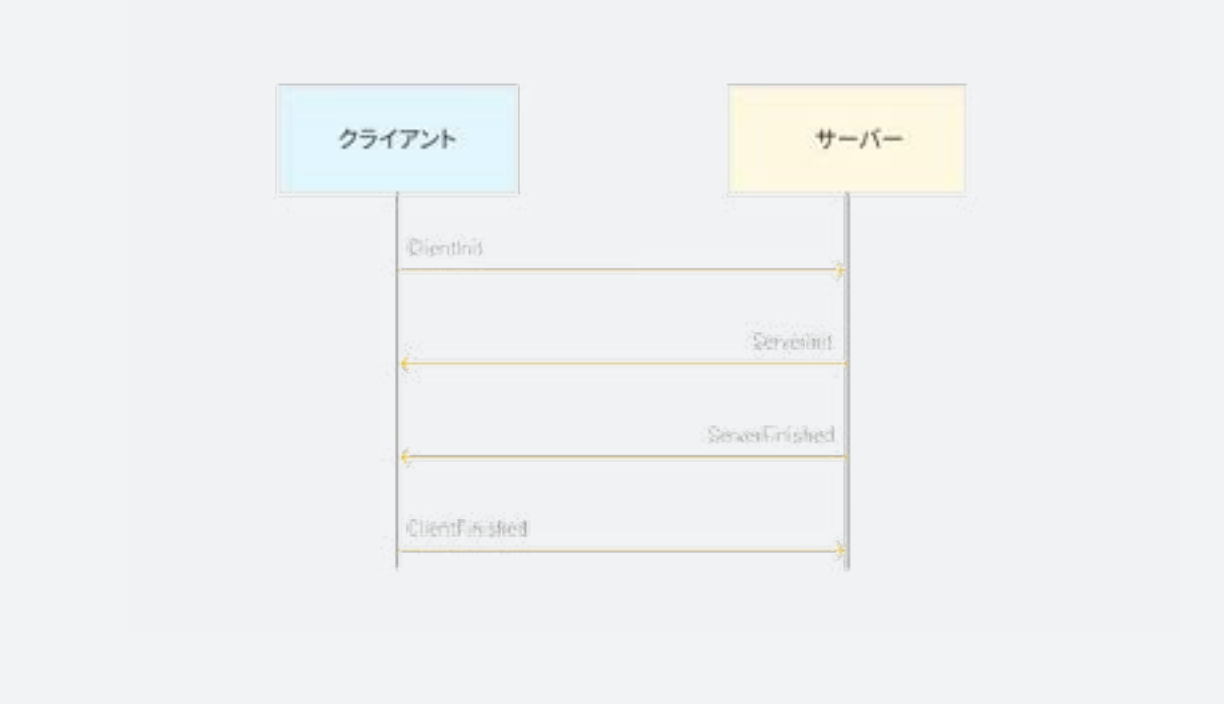
4.1 handshake プロトコル

ALTS handshake プロトコルは、Perfect Forward Secrecy (PFS) とセッション再開の両方をサポートする、Diffie-Hellman ベースの認証鍵交換プロトコルです。ALTS のインフラストラクチャでは、各クライアントおよびサーバーがそれぞれの ID と、信頼される署名サービス検証鍵にチェーンした楕円曲線 Diffie-Hellman (ECDH) 鍵を含む証明書を持っています。ALTS では handshake で PFS が使用されない場合でも、これらの静的 ECDH 鍵が頻繁に更新されて前方秘匿性が更新されるため、PFS はデフォルトでは有効になっていません。handshake の際、クライアントとサーバーは共有の転送暗号鍵と、その暗号鍵を使用して保護されるレコード プロトコルを安全にネゴシエートします。たとえばクライアントとサーバーは、AES-GCM を使用した RPC セッションを保護するための 128 ビットキーに同意する場合があります。handshake は、4 つのシリアル化されたプロトコル バッファ メッセージで構成されています。その概要を図 6 に示します。

1. クライアントが `ClientInit` メッセージを送信して handshake を開始します。このメッセージには、クライアントの handshake 証明書と、クライアントがサポートする handshake 関連の暗号とレコード プロトコルのリストが含まれています。クライアントが終了済みのセッションを再開しようとしている場合は、再開識別子と暗号化されたサーバー再開チケットが含まれます。

図 6:

ALTS handshake プロトコル メッセージ



2. サーバーは `ClientInit` メッセージを受信すると、クライアントの証明書を検証します。それが有効であれば、サーバーはクライアントから提供されたリストから、handshake 暗号と記録プロトコルを選択します。サーバーは `ClientInit` メッセージに含まれる情報と自身のローカル情報を組み合わせて、DH 交換の結果を計算します。この結果は、以下のセッション シークレットを生成するプロトコルの写しとともに、鍵導出関数[13] への入力として使用されます。

- ペイロード メッセージを暗号化し認証するために使用される**レコード プロトコル**秘密鍵 M。
- 将来のセッションで再開チケットに使用される**再開シークレット** R。
- **認証シークレット** A。

サーバーはその証明書、選択した handshake 暗号、レコード プロトコル、および必要に応じて暗号化された再開チケットを含む、 `ServerInit` メッセージを送信します。

3. サーバーが、handshake オーセンティケータを含む `ServerFinished` メッセージを送信します[14] 。このオーセンティケータの値は、事前定義によるビット文字列とオーセンティケータのシークレット A によって算出された、ハッシュベースのメッセージ認証コード（HMAC）を使用して計算されます。

4. クライアントは `ServerInit` を受信すると、サーバー証明書を検証し、サーバーと同様の DH 交換結果を計算して、同じ M、R、A シークレットを取得します。クライアントは取得した A を使用して、受信した `ServerFinished` メッセージのオーセンティケータ値を検証します。handshake 処理のこの時点で、クライアントは M を使用したメッセージの暗号化を開始できます。これでクライアントが暗号化されたメッセージを送信できるようになるため、ALTS には RTT handshake プロトコルが 1 つあることになります。

[13] 具体的には、RFC-5869 で定義された HKDF-Extract および HKDF-Expand です。
[14] ALTS の handshake プロトコル実装では、ServerInit メッセージと ServerFinished メッセージが 1 つのワイヤ ペイロードに連結されます。

5. handshake の最後にクライアントは、異なる事前定義のビット文字列を使用して計算された同様のオーセンティケータ値（手順 3 参照）を含む `ClientFinished` メッセージを送信します。必要に応じて、クライアントは将来のセッション用に、暗号化された再開チケットを含めることができます。このメッセージがサーバーによって受信され検証されると、ALTS handshake プロトコルが終了し、サーバーは M を使用して以降のペイロード メッセージの暗号化と認証を開始することができます。

handshake プロトコルは、Google の社内セキュリティ分析チームの Thai Duong がレビューを行い、Bruno Blanchet が Martin Abadi の助力を得て、Proverif[15] ツールを使用しながら正式に検証しました。

4.2 レコード プロトコル

handshake プロトコルのセクションでは、handshake プロトコルを使用してレコード プロトコルのシークレットがネゴシエートされる方法を説明しました。このプロトコル シークレットは、ネットワーク トラフィックの暗号化と認証に使用されます。これらのオペレーションを実行するスタックのレイヤは、ALTS レコード プロトコル（ALTSRP）と呼ばれます。

ALTSRP には、さまざまな鍵サイズとセキュリティ機能を備えた一連の暗号化スキームが含まれています。handshake の際、クライアントは優先順位別にソートされた優先スキームのリストを送信します。サーバーはクライアントのリストから、サーバーのローカル構成と一致する最初のプロトコルを選択します。このスキーム選択の方法により、クライアントとサーバーの両者が異なる暗号化の優先順位を持つことができるため、Google では暗号化スキームを段階的に導入（または削除）することが可能になっています 。

4.2.1 フレーム処理

フレームは ALTS における最小のデータ単位です。各 ALTSRP メッセージはそのサイズに応じて、1 つまたは複数のフレームで構成できます。各フレームには以下のフィールドが含まれます。

- **Length:** フレームの長さ（バイト単位）を示す、符号なしの 32 ビット値。この長さ 4 バイトのフィールドは、全フレーム長の一部には含まれません。
- **Type:** フレームのタイプ（例: データフレームなど）を指定する 32 ビット値。
- **Payload:** 実際に認証され、必要に応じて暗号化された送信データ。

[15] ProVerif: 形式モデルにおける暗号プロトコル検証ツール。

フレームの最大長は 1 MB + 4 バイトです。現行の RPC プロトコルでは、フレーム長をさらに制限しています。短いフレームのほうがバッファリングに必要なメモリが少ないからです。大きなフレームは、サーバーを枯渇させようとするサービス拒否 (DoS) 攻撃の際、潜在的な攻撃者によって利用されるおそれもあります。Google ではフレーム長を制限するだけでなく、同じレコード プロトコルのシークレット M を使用して暗号化できるフレームの数も制限しています。この制限は、フレーム ペイロードの暗号化と復号化に使用される暗号化スキームによって異なります。この制限に達した場合は、接続を閉じる必要があります。

4.2.2 ペイロード

ALTS では各フレームに、完全性が保護され必要に応じて暗号化されたペイロードが含まれています[16]。本ドキュメントの発行時点で、ALTS は以下のモードをサポートしています。

- AES-128-GCM、AES-128-VCM: それぞれ AES-GCM モードおよび AES-VCM モード、128 ビットキー使用。これらのモードではそれぞれ GCM および VCM 方式[17]を使用して、ペイロードの機密性と完全性が保護されます。

- AES-128-GMAC、AES-128-VMAC: これらのモードはタグ コンピューティングについて、それぞれ GMAC と VMAC を使用した完全性のみの保護をサポートしています。ペイロードは、その完全性を保護する暗号化タグを含むプレーン テキストで転送されます。

Google では脅威のモデルおよびパフォーマンス要件によって、さまざまな保護モードを使用しています。通信するエンティティが Google によってまたは Google のために管理されている同じ物理的境界内にある場合は、完全性のみの保護が使用されます。これらのエンティティはデータの機密性に基づいて、選択により認証付き暗号にアップグレードすることも可能です。通信しているエンティティが、Google によってまたは Google のために管理されている別の物理的境界内にあり、通信が広域ネットワークを通過する場合は、選択されたモードに関係なく、接続のセキュリティが認証付き暗号に自動的にアップグレードされます。Google では、データが Google によってまたは Google のために管理されている物理的境界の外に転送される場合、同じ厳格なセキュリティ手段を適用することができないため、転送中のデータにさまざまな保護を適用しています。

[16] ペイロードの暗号化は、handshake でレコード プロトコルの一部としてネゴシエートされます。
[17] 128 ビットの AES-GCM スキームは NIST 800-38D に基づいており、AES-c については AES-VCM: 整数ベースのユニバーサル ハッシュ関数を使用した AES-GCM の構築で詳細に説明されています。

各フレームはそれぞれ別個に完全性が保護され、必要に応じて暗号化されます。両方のピアでリクエスト カウンタとレスポンス カウンタの両方が維持され、これらは通常のオペレーション中に同期が行われます。サーバーが順序どおりでないリクエストや繰り返しのリクエストを受信すると、暗号の完全性検証が失敗し、リクエストは破棄されます。同様にクライアントでも、繰り返されるレスポンスや順序の誤ったレスポンスは破棄されます。さらに両方のピアが（フレーム ヘッダーに値を含めるのではなく）カウンタを維持していることで、転送時のバイトがさらに節約されます。

4.3 セッションの再開

ALTS では、ユーザーは負荷の高い非対称暗号化のオペレーションを行うことなく、以前のセッションを再開できます。セッション再開は ALTS の Handshake プロトコルに組み込まれている機能です。

ALTS handshake により、クライアントとサーバーは再開チケットを安全に交換（およびキャッシュ）できます。再開チケットはその後、接続を再開するために使用できます[18]。キャッシュされたそれぞれの再開チケットは、同一のデータセンター セル内で同じ ID で実行されているすべてのワークロードに固有の再開識別子（ID_R）によりインデックス付けされます。これらのチケットは、対応する識別子に関連付けられた対称鍵を使用して暗号化されます。

ALTS handshake により、クライアントとサーバーは再開チケットを安全に交換できます。再開チケットはその後、接続を再開するために使用できます。

ALTS は 2 種類のセッション再開をサポートしています。

1. **サーバー側のセッション再開:** クライアントが、サーバー ID と取得した再開シークレット R を含む再開チケットを作成し暗号化します。再開チケットは handshake 終了時に、`ClientFinished` メッセージでサーバーに送信されます。将来のセッションでは、サーバーが `ServerInit` メッセージでチケットをクライアントに送り返すことで、セッションを再開できます。チケットを受信したクライアントは、再開シークレット R とサーバーの ID の両方を復元できます。クライアントはこの情報を使用してセッションを再開することができます。

   ID_R は常に ID と関連付けられており、特定の接続に関連付けられるものではありません。ALTS では、複数のクライアントが同じデータセンター内で同じ ID を使用できます。これにより、ク

[18] セッションの再開では、一時的パラメータが含まれない場合にのみ軽量な対称オペレーションが行われます。

ライアントは以前に通信していない可能性のあるサーバーとのセッションを再開できます。たとえば、ロードバランサが同じアプリケーションを実行している別のサーバーにクライアントを送信した場合などです。

2. クライアント側のセッション再開: handshake の終了時、サーバーは暗号化された再開チケットを、`ServerFinished` メッセージでクライアントに送信します。このチケットには再開シークレット R とクライアントの ID が含まれています。クライアントはこのチケットを使用して、同じ ID_R を共有する任意のサーバーとの接続を再開できます。

セッションが再開されると、再開シークレット R を使用して新しいセッション シークレット M'、R'、A' が導出されます。M' はペイロード メッセージの暗号化と認証に、A' は `ServerFinished` および `Client Finished` メッセージの認証に使用され、R' は新しい再開チケット内にカプセル化されます。同じ再開シークレット R が 2 回以上は使用されないことに注意してください

5. トレードオフ

5.1 KCI (Key Compromise Impersonation) 攻撃

ALTS handshake プロトコルは設計上、KCI (Key Compromise Impersonation) 攻撃の影響を受けやすくなっています。敵対者がワークロードの DH 秘密鍵または再開鍵を不正に入手した場合は、その鍵を使用してこのワークロードに対し他のワークロードを偽装できます[19]。これは

[19] 鍵共有プロトコルとそのセキュリティ分析。

Google の再開脅威モデルに明示的に含まれています。Google ではある ID の 1 つのインスタンスが発行した再開チケットを、その ID の他のインスタンスでも使用できるようにしたいと考えているからです。

ALTS handshake プロトコルには KCI 攻撃からの保護を実現するバリアントがありますが、これは再開が望まれない環境でのみ使用する価値があるものです。

5.2 handshake メッセージのプライバシー

ALTS はどの内部 ID が通信を行っているかを隠すようには設計されていないため、ピアの ID を隠すための handshake メッセージの暗号化は行われません。

5.3 Perfect Forward Secrecy（前方秘匿性）

ALTS では、Perfect Forward Secrecy（PFS、前方秘匿性）はサポートされていますが、デフォルトで有効にはなっていません。代わりに大半のアプリケーションについては、頻繁な証明書ローテーションを使用して前方秘匿性が確立されます。TLS1.2（およびそれ以前のバージョン）では、セッション再開は PFS で保護されていません。ALTS で PFS を有効にすると、PFS は再開されたセッションに対しても有効になります。

5.4 ゼロ ラウンドトリップ再開

TLS 1.3 ではゼロ ラウンドトリップ（0-RTT）を必要とするセッション再開が可能になっていますが、これはセキュリティ特性が弱くなっています[20]。Google の RPC 接続は一般的に長時間になるため、Google では ALTS に 0-RTT のオプションを含めていません。この結果、チャネル設定のレイテンシを低減させることは、0-RTT handshake によって必要となる複雑さの増大やセキュリティ低下との適切なトレードオフにはなりませんでした。

6. その他のリファレンス

Googleで転送中のデータがどのように暗号化されるかについては、Google Cloud での転送データの暗号化のホワイトペーパーをご覧ください。

セキュリティが Google の技術インフラにどのように組み込まれているかの概要は、 Google インフラストラクチャのセキュリティ設計の概要をご覧ください。

[20] ゼロ ラウンドトリップ時間のリプレイ攻撃: TLS 1.3 handshake 候補のケース