AWS Key Management Service のベストプラクティス

2017 年 *4* 月

注意

本書は、情報提供の目的のみのために提供されるものです。本書の発行時点における AWS の現行製品と慣行を表したものであり、それらは予告なく変更されることがあります。お客様は本書の情報および AWS 製品の使用について独自に評価する責任を負うものとします。これらの情報は、明示または黙示を問わずいかなる保証も伴うことなく、「現状のまま」提供されるものです。本書のいかなる内容も、AWS、その関係者、サプライヤー、またはライセンサーからの保証、表明、契約的責任、条件や確約を意味するものではありません。お客様に対する AWS の責任は、AWS 契約により規定されます。本書は、AWS とお客様の間で行われるいかなる契約の一部でもなく、そのような契約の内容を変更するものでもありません。

目次

序文 1

ID およびアクセス管理 2

AWS KMS および IAM ポリシー 2

キーポリシー 2

アカウント間でのキーの共有 5

CMK 付与 5

暗号化コンテキスト 6

多要素認証 7

発見的統制 8

CMK の監査 8

CMK の使用の検証 8

インフラストラクチャのセキュリティ 9

カスタマーマスターキー 9

大規模環境での AWS KMS の使用 12

データ保護 13

AWS KMS の一般的なユースケース 13

AWS サービスでの保存データ暗号化の実施 15

インシデント対応 17

AWS KMS のセキュリティの自動化 17

CMK の削除と無効化 17

まとめ 18

寄稿者 19

文書の改訂 19

要約

AWS Key Management Service (AWS KMS) はマネージド型サービスであり、アマゾン ウェブ サービス (AWS) が基盤となるインフラストラクチャの可用性、物理的セキュリティ、論理的アクセスコントロール、メンテナンスを管理するので、お客様はアプリケーションの暗号化ニーズに集中することができます。さらに、AWS KMS は、行われたすべての API コールのログを提供することでキーの使用状況の監査を可能にし、コンプライアンスや規制要件を満たすのに役立ちます。

顧客は、自社の環境で AWS KMS を効果的に実装する方法を知る必要があります。このホワイトペーパーでは、各種カスタマーマスターキーの違い、最少の権限を守るための AWS KMS キーポリシーの使用、キーの使用の監査、AWS における機密情報保護のユースケースの紹介を含めて、「AWS クラウド導入フレームワーク: セキュリティの観点」のホワイトペーパーで説明している機能ごとに AWS KMS を使用する方法を説明します。

序文

AWS Key Management Service (AWS KMS) は、マネージド型サービスであり、データの暗号化に使用する暗号化キーを簡単に作成して管理できます。AWS KMS は、ハードウェアセキュリティモジュール (HSM) を使用してキーのセキュリティを保護します。[1]AWS KMS を使用して、AWS サービスやアプリケーションでデータを保護することができます。AWS Key Management Service の暗号化の詳細のホワイトペーパーは、データのセキュリティとプライバシーを確保するためにサービス内で実装されている設計と制御を説明しています。[2]

AWS クラウド導入フレームワーク (CAF) のホワイトペーパーは、クラウドコンピューティングへ移行している組織のさまざまな部分を調整するためのガイダンスを提供します。[3]AWS CAF のガイダンスは、観点と呼ばれる、クラウドベースの IT システムの実装に関連する重点領域に分割されます。CAF セキュリティの観点のホワイトペーパーは、次の 5 つの主要な機能を通じて組織のセキュリティの変換を促進するのに役立つ原則を整理しています。ID およびアクセス管理、発見的統制、インフラストラクチャのセキュリティ、データ保護、インシデント対応です。[4]

CAF セキュリティの観点の各機能について、このホワイトペーパーは、さまざまなユースケースや進捗状況を測定する手段で機密情報を保護するために組織がどのように AWS KMS を使用すればよいかを詳しく説明しています。

- **ID およびアクセス管理:** 複数のアクセスコントロールのメカニズムを作成し、それぞれに対してアクセス許可を管理できるようにします。
- **発見的統制:** ネイティブなロギングの機能と、サービスへの可視性を提供します。
- **インフラストラクチャのセキュリティ:** 要件に適合するようにセキュリティを制御する機能を提供します。
- **データ保護:** データの可視性と制御を維持できる機能を提供します。
- **インシデント対応:** 有害なインシデントに対応し、それらを管理、削減し、インシデント中やインシデント後に操作を復元する機能を提供します。

ID およびアクセス管理

ID およびアクセス管理の機能は、AWS KMS 内のアクセス管理のコントロールを決定し、確立されたベストプラクティスや内部ポリシーに従ってインフラストラクチャを保護するためのガイダンスを提供します。

AWS KMS および IAM ポリシー

AWS Identity and Access Management (IAM) のポリシーをキーポリシーと組み合わせて使用し、AWS KMS でカスタマーマスターキー (CMK) へのアクセスをコントロールすることができます。このセクションでは、AWS KMS の状況での IAM の使用について説明します。IAM サービスに関する詳細情報は提供しません。IAM の完全な ドキュメントについては、AWS IAM ユーザーガイドを参照してください。[5]

IAM アイデンティティ (つまり、ユーザー、グループ、ロール) にアタッチされたポリシーは、アイデンティティベースのポリシー (または *IAM* ポリシー) と呼ばれます。IAM の外部のリソースにアタッチされたポリシーは、リソースベースのポリシーと呼ばれます。AWS KMS では、リソースベースのポリシーをカスタマーマスターキー (CMK) にアタッチする必要があります。これらを、キーポリシーと呼びます。すべての KMS CMK にキーポリシーがあり、それを使用して CMK へのアクセスをコントロールする必要があります。IAM ポリシー単独では、CMK へのアクセスを許可するのに十分ではありませんが、CMK のキーポリシーと組み合わせて使用することはできます。そのために、CMK のキーポリシーに IAM ポリシーを有効にするポリシーステートメントが含まれていることを確認します。[6]

アイデンティティベースの IAM ポリシーを使用することで、AWS アカウント内での KMS API コールにより詳細なアクセス権を付与することで、最少の権限を実現できます。IAM ポリシーは、アクションを実行するプリンシパルに明示的にアクセス許可を付与しない限り、デフォルトで拒否のポリシーに基づくことを忘れないでください。

キーポリシー

キーポリシーは、AWS KMS で CMK へのアクセスをコントロールする主要な方法です。それぞれの CMK には、キーの使用および管理に関するアクセス許可を定義するキーポリシーがアタッチされています。デフォルトのポリシーによって、定義するプリンシパルが有効になり、アカウントの root ユーザーは、キーを参照する IAM ポリシーを追加することができます。デフォルトの CMK ポリシーを編集して、最少の権限に関する組織のベストプラクティスと合致させることをお勧めします。暗号化されたリソースにアクセスするには、プリンシパルがリソースを使用するためのアクセス許可、ならびにリソースを保

護する暗号化キーを使用するためのアクセス許可を持つ必要があります。プリンシパルがこれらのアクションのいずれかに必要なアクセス許可を持っていない場合、暗号化されたリソースを使用するリクエストは拒否されます。

CMK キーポリシー内での `kms:ViaService` 条件ステートメントの使用によって、特定の AWS サービスによってだけ使用されるように、CMK を制限することもできできます。詳細については、AWS KMS 開発者ガイドを参照してください。[7]

暗号化された Amazon Elastic Block Store (EBS) ボリュームを作成して使用するには、Amazon EBS を使用するためのアクセス許可が必要です。CMK に関連するキーポリシーは、次のようなコードを含んでいる必要があります。

```
{
   "Sid": "Allow for use of this Key",
   "Effect": "Allow",
   "Principal": {
     "AWS": "arn:aws:iam::111122223333:role/UserRole"
   },
   "Action": [
     "kms:GenerateDataKeyWithoutPlaintext",
     "kms:Decrypt"
   ],
   "Resource": "*"
 },
 {
   "Sid": "Allow for EC2 Use",
   "Effect": "Allow",
   "Principal": {
     "AWS": "arn:aws:iam::111122223333:role/UserRole"
   },
   "Action": [
     "kms:CreateGrant",
     "kms:ListGrants",
     "kms:RevokeGrant"
   ],
   "Resource": "*",
   "Condition": {
     "StringEquals": {
       "kms:ViaService": "ec2.us-west-2.amazonaws.com"
     }
   }
 }
```

この CMK ポリシーでは、最初のステートメントが、データキーを生成し、必要に応じて CMK からデータキーを復号化する機能の具体的な IAM のプリンシパルを提供しています。これらの 2 つの API は、Amazon Elastic Compute Cloud (EC2) インスタンスにアタッチされている EBS ボリュームを暗号化するのに必要です。

このポリシーの 2 番目のステートメントは、Amazon EC2 に対する付与を作成、一覧、取り消しできる機能の具体的な IAM のプリンシパルを提供しています。付与は、キーを自身のために使用できるように、アクセス許可の一部を AWS サービスまたは他のプリンシパルに委任するために使用されます。この例では、条件ポリシーが明示的に、Amazon EC2 だけが付与を使用できることを保証しています。Amazon EC2 は、計画的または計画外の停電のためにボリュームが切り離された場合に、暗号化された EBS ボリュームをインスタンスに再アタッチするためにそれらを使用します。これらのイベントが発生した場合、その時点で監査のために AWS CloudTrail に記録されます。

CMK ポリシーを開発する場合、AWS 内でどのように[ポリシーステートメントが評価されるか]に留意してください。これは、[CMK へのアクセスを制御するために IAM を有効にした]場合、許可されたアクションが許可されるか拒否されるかを AWS が評価するときに、CMK ポリシーが IAM ポリシーと結合されることを意味します。さらに、キーの使用と管理が必要な当事者に限定されていることを確認する必要があります。

最小の権限/職務分掌

キーポリシーは、CMK へのアクセスを付与するリソース、アクション、効果、プリンシパル、条件を指定します。キーポリシーを使用すると、CMK へのアクセス許可をより細かくプッシュして、最小の権限を実現することができます。たとえば、アプリケーションが KMS API コールを使用してデータを暗号化することがありますが、同じアプリケーションがデータを復号化するユースケースはありません。このユースケースでは、キーポリシーは *kms:Encrypt* アクションへのアクセスは付与できますが、*kms:Decrypt* へは付与できず、露出の可能性を減少させます。さらに、AWS では、キーに関連付けられている管理の権限から、使用の権限を分離できます。これは、キーポリシーを操作できるユーザーでも、そのキーを暗号機能に使用するアクセス許可は持たない場合があることを意味します。

機密情報を保護するために CMK が使用されている場合、対応するキーポリシーが最少の権限のモデルに従っていることを確認するように作業する必要があります。これには、IAM ポリシーに kms:* 権限を**含めない**ようにすることが含まれます。このポリシーは、プリンシパルがアクセスできるすべての CMK に対しする管理権限と使用権限の両方をプリンシパルに付与します。同様に、キーポリシー内のプリンシパルに kms:* 権限を含めると、CMK の管理権限と使用権限の両方を与えることになります。

明示的な拒否ポリシーが暗黙の拒否ポリシーよりも優先されることを覚えておくことが重要です。"Effect: Deny" と同じポリシーステートメントで NotPrincipal を使用すると、ポリシーステートメントで指定されている権限は、指定されたプリンシパルを除く、すべてのプリンシパルに対して明示的に拒否されます。最上位の KMS ポリシーは、実際にそれらを必要とするロールを除いて、実質的にすべての KMS オペレーションに対するアクセスを明示的に拒否することができます。この手法は、権限のないユーザーが KMS へのアクセス許可を自身に付与するのを防ぐのに役立ちます。

アカウント間でのキーの共有

CMK キーポリシー内に信頼できるアカウントのルートプリンシパルを含めると、AWS KMS 内の CMK への権限の委任が発生します。さらに、信頼できるアカウントは、IAM ポリシーを使用して、これらの権限を IAM ユーザーや自身のアカウントのロールに委任することができます。この方法でキーポリシーの管理を簡素化できる可能性がありますが、同時に、委任された権限が正しく管理されていることを確認するのは信頼できるアカウントに依存します。もう 1 つの方法として、KMS キーポリシーのみを使用してすべての許可されたユーザーへの権限を管理する方法もありますが、これはキーポリシーを複雑にして、管理しやすさが低減します。どのような方法を採用するかに関わらず、確実に最少の権限のモデルに従うには、特定の信頼をキーごとに分割する必要があります。

CMK 付与

キーポリシーの変更は、AWS の他の場所でのポリシー編集に使用されているのと同じ権限モデルに従います。つまり、ユーザーはキーポリシーを変更する権限を持っているか、持っていないかのいずれかになります。CMK の `PutKeyPolicy` 権限を持つユーザーは、CMK のキーポリシーを、選択した別のキーで完全に置き換えることができます。キーポリシーを使用すると、他のプリンシパルが CMK へアクセスすることを許可できますが、キーポリシーが最適に機能するのは、権限を比較的静的に割り当てる場合です。より詳細なアクセス権限の管理を可能にするには、付与を使用すします。付与は、直接の API コールがない場合に、他のプリンシパルが CMK を使用するためのスコープダウンされた一時的な権限を定義する場合に便利です。

キーへのアクセスを制御するために付与を使用するアプリケーションを設計する場合は、キーごとの付与と、キー制限ごとのプリンシパルの付与を認識することが重要です。これらの制限に達するのを避けるために使用された後、廃止されたプリンシパルが付与しないことを確認します。

暗号化コンテキスト

AWS KMS API への権限を制限することに加えて、AWS KMS は、暗号化コンテキストを使用して KMS API コールに追加の認証レイヤーを追加することもできます。暗号化コンテキストとは、AWS KMS で保護された情報と関連付ける必要がある追加データのキーと値のペアです。これは、AWS KMS で暗号化された暗号テキストで、認証された暗号化の追加認証データ (AAD) に組み込まれます。暗号化操作で暗号化コンテキストの値を送信する場合は、対応する復号化操作で転送する必要があります。ポリシー内で暗号化コンテキストを使用すると、暗号化されたリソースをより厳重にコントロールできます。暗号化コンテキストは CloudTrail に記録されるため、監査の観点からキーの使用状況をより詳細に把握できます。暗号化コンテキストは暗号化されておらず、CloudTrail ログで表示されることに注意してください。暗号化コンテキストは機密情報と見なされるべきではなく、機密性は不要なはずです。

AWS KMS を使用する AWS サービスは、暗号化コンテキストを使用してキーの範囲を制限します。たとえば、Amazon EBS はボリュームを暗号化/復号化する際に暗号化コンテキストとしてボリューム ID を送信し、スナップショットを取得するとスナップショット ID をコンテキストとして使用します。Amazon EBS がこの暗号化コンテキストを使用しなかった場合、EC2 インスタンスはその特定の CMK のもとで EBS ボリュームを復号化することができます。

また暗号化コンテキストは、開発するカスタムアプリケーションにも使用でき、暗号化コンテキストが暗号化呼び出しで渡されたものと一致する場合にのみ、復号化呼び出しが成功するようにすることで制御の追加レイヤーとして機能します。特定のアプリケーションの暗号化コンテキストが変更されない場合は、そのコンテキストを条件ステートメントとして AWS KMS キーポリシーに含めることができます。たとえば、データの暗号化と復号化を必要とするアプリケーションがある場合、期待値を提供するように CMK でキーポリシーを作成することができます。次のポリシーでは、アプリケーション名「ExampleApp」とその現在のバージョン「1.0.24」が、暗号化および復号化呼び出し中に AWS KMS に渡される値であることを確認しています。異なる値が渡された場合、呼び出しは拒否され、復号化または暗号化のアクションは実行されません。

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/RoleForExampleApp"
```

```
  },
  "Action": [
    "kms:Encrypt",
    "kms:Decrypt"
  ],
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "kms:EncryptionContext:AppName": "ExampleApp",
      "kms:EncryptionContext:Version": "1.0.24"
    }
  }
}
```

この暗号化コンテキストの使用は、許可された当事者および/またはアプリケーションだけが CMK にアクセスして使用することをさらに確実にするのに役立ちます。これで当事者は、AWS KMS への IAM 権限、つまり要求された方法でキーを使用できるようにする CMK ポリシーを必要とし、最終的には予想される暗号化コンテキストの値を知る必要があります。

多要素認証

特定のアクションに対して追加のセキュリティレイヤーを提供するには、重要な KMS API コールで多要素認証 (MFA) を使用して追加の保護レイヤーを実装することができます。これらの呼び出しは、`PutKeyPolicy`、`ScheduleKeyDeletion`、`DeleteAlias`、`DeleteImportedKeyMaterial` などです。これは、MFA デバイスがいつ認証の一部として使用されたかをチェックするキーポリシー内の条件ステートメントによって実行できます。

誰かが重大な AWS KMS アクションの 1 つを実行しようとすると、以下の CMK ポリシーが、アクションを実行する前に、その MFA が最近 300 秒または 5 分以内に認証されたことを検証します。

```
{
  "Sid": "MFACriticalKMSEvents",
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:user/ExampleUser"
  },
  "Action": [
```

```
      "kms:DeleteAlias",
      "kms:DeleteImportedKeyMaterial",
      "kms:PutKeyPolicy",
      "kms:ScheduleKeyDeletion"
    ],
    "Resource": "*",
    "Condition":{
      " NumericLessThan ":{"aws: MultiFactorAuthAge":"300"}
    }
  }
```

発見的統制

発見的統制機能は、顧客の環境をよりよく把握するために必要な情報をログに記録するように AWS KMS を適切に設定することを保証します。

CMK の監査

AWS KMS は、CloudTrail と統合されています。AWS KMS でキーの使用状況を監査するには、AWS アカウントで CloudTrail のログを有効にする必要があります。これにより、AWS アカウントのキーで行われたすべての KMS API コールが自動的にファイルに記録され、指定した Amazon Simple Storage Service (S3) バケットに配信されます。CloudTrail によって収集された情報を使用すると、どのようなリクエストが行われたか、リクエストが行われたソース IP アドレス、リクエストの実行者、リクエストの実行日時などを判断できます。

AWS KMS は他の多くの AWS サービスとネイティブで統合されており、モニタリングを容易にします。これらの AWS サービス、または既存のセキュリティツールスイートを使用して、KMS キーで `ScheduleKeyDeletion, PutKeyPolicy, DeleteAlias, DisableKey, DeleteImportedKeyMaterial` などの特定のアクションの CloudTrail ログを監視することができます。さらに、AWS KMS は、CMK がローテーション、削除されたり、CMK でインポートされたキーマテリアルが期限切れになったりすると、Amazon CloudWatch イベントを発行します。

CMK の使用の検証

キーの管理と使用に関連する監査データのキャプチャに加えて、確認しているデータが、確立されているベストプラクティスおよびポリシーに沿ったものであることを確認する必要があります。1 つの方法は、

CloudTrail のログを継続的に監視し、確認することです。もう 1 つの方法は、AWS Config ルールを使用することです。AWS Config ルールを使用することで、多くの AWS サービスが適切に設定されていることを確認できます。たとえば、EBS ボリュームで、AWS Config ルールの `ENCRYPTED_VOLUMES` を使用して、アタッチされている EBS ボリュームが暗号化されていることを確認できます。

キーのタグ

CMK には、さまざまな目的でタグが適用されています。最も一般的な使用方法は、特定の CMK をビジネスカテゴリ (コストセンター、アプリケーション名、所有者など) に戻すことです。さらにタグを使用して、特定のアクションに正しい CMK が使用されていることを確認できます。たとえば、CloudTrail のログで、特定の KMS アクションに対して、使用されている CMK が、使用されているリソースと同じビジネスカテゴリに属していることを確認できます。これまでは、リソースカタログ内での検索が必要になることがありましたが、現在は AWS KMS や他の多くの AWS サービスでタグ付けされているため、この外部参照は不要です。

インフラストラクチャのセキュリティ

インフラストラクチャのセキュリティ機能は、重要な情報を保護しながらビジネスに合わせて拡張できる俊敏な実装を確実に実現するための AWS KMS の設定方法に関するベストプラクティスを提供します。

カスタマーマスターキー

AWS KMS では、キー階層は CMK で始まります。CMK を使用して最大 4 KB のデータブロックを直接暗号化することができます。また、任意のサイズの基盤となるデータを保護するデータキーを保護するために使用することもできます。

AWS マネージド型およびカスタマーマネージド型の CMK

CMK は、次の 2 つの一般的なタイプに分類できます。AWS マネージド型およびカスタマーマネージド型です。そのサービスの AWS マネージド型 CMK のもとで AWS リソースのサーバーサイド暗号化を有効にすることを初めて選択した場合に、AWS マネージド型 CMK が作成されます (例、SSE-KMS)。AWS マネージド型 CMK は、AWS アカウントとそれが使用されるリージョンに固有です。AWS マネージド型 CMK は、作成された特定の AWS サービス内のリソースを保護する目的でのみ使用できます。カスタマーマネージド型 CMK が提供する細かいレベルのコントロールは提供されません。より細かいレベルのコントロールのためのベストプラクティスは、サポートされているすべての AWS サービスとアプリケーシ

ョンでカスタマーマネージド型 CMK を使用することです。要求に応じてカスタマーマネージド型 CMK を作成すると、明示的なユースケースに基づいて設定する必要があります。

次の表は、AWS マネージド型 CMK とカスタマーマネージド型 CMK の主な相違点と類似点をまとめたものです。

	AWS マネージド型 CMK	カスタマーマネージド型 CMK
作成	顧客のために AWS が生成する	顧客が生成する
ローテーション	3 年に 1 回自動的	1 年に 1 回自動的にオプトインで、またはオンデマンドで手動で
削除	削除できません	削除できます
使用範囲	特定の AWS サービスに制限	KMS/IAM ポリシーで管理
キーアクセスポリシー	AWS マネージ	カスタマーマネージ
ユーザーアクセス管理	IAM ポリシー	IAM ポリシー

カスタマーマネージド型 CMK の場合、基盤となるキーマテリアルの作成に 2 つのオプションがあります。AWS KMS を使用して CMK を作成する場合、KMS に暗号材料を作成させるか、独自のキーマテリアルをインポートすることができます。これらのオプションはどちらも、環境内で CMK を使用する場合と同じレベルのコントロールと監査を提供します。独自の暗号マテリアルをインポートする機能では、以下のことが可能です。

- ランダム性要件を満たす承認されたソースを使用してキーマテリアルを生成したことを証明する。
- AWS サービスで独自のインフラストラクチャのキーマテリアルを使用し、AWS KMS を使用して AWS 内でのそのキーマテリアルのライフサイクルを管理する。
- AWS でキーマテリアルの有効期限を設定して手動で削除するが、将来再び有効にすることができる機能を得る。
- キーマテリアルのオリジナルコピーを所有し、キーマテリアルの完全なライフサイクルにわたって耐久性と災害復旧のために、AWS の外で保管する。

インポートされたキーマテリアルを使用するか、KMS によって生成されたキーマテリアルを使用するかの決定は、組織のポリシーおよびコンプライアンス要件によって異なります。

キーの作成と管理

AWS は AWS KMS を使用してキーの作成および管理を簡単にできるため、個々のキーの影響範囲を最適にコントロールできるようにサービスの使用方法を計画することをお勧めします。以前は、異なる地理的地域、環境、時にはアプリケーションでも同じキーを使用していた可能性があるかもしれません。AWS KMS では、データ分類レベルを定義し、レベルごとに少なくとも 1 つの CMK を設定する必要があります。たとえば、「機密」と分類されたデータに対して CMK を定義するなどです。こうすることで、許可されたユーザーだけが、自分の仕事を完了するために必要なキーマテリアルに対する権限を与えられます。

また、AWS KMS の使用をどのように管理するかを決定する必要があります。機密データの暗号化と復号化の機能を必要とする各アカウント内で KMS キーを作成することが、大半の顧客にとって最適ですが、いくつかの集中アカウントから CMK を共有するという選択肢もあります。大部分のインフラストラクチャと同様に同じアカウントで CMK を管理することは、ユーザーがこれらのキーを使用する AWS サービスをプロビジョニングして実行するのに役立ちます。検索を行うプリンシパルが外部アカウントが所有するリソースに対して明示的な List* 権限を持っていない限り、AWS サービスはクロスアカウント検索を許可しません。これは、サービスコンソールベースの検索ではなく、CLI または SDK を介してのみ実行することもできます。さらに、認証情報をローカルアカウントに保存することにより、特定の CMK へのアクセスを必要とする IAM プリンシパルを知っている個人に権限を委任する方が簡単かもしれません。集中化モデルを使用してキーを共有していた場合、AWS KMS 管理者は最少の権限を確保するために CMK のすべてのユーザーに対して完全な Amazon リソースネーム (ARN) を知る必要があります。そうしないと、管理者がキーに過度に大きな権限を与えてしまう可能性があります。

組織は、CMK のローテーションの頻度も考慮する必要があります。多くの組織は、毎年 CMK をローテーションします。KMS により生成されたキーマテリアルのカスタマーマネージド型 CMK の場合、これは簡単に適用できます。CMK の年間ローテーションスケジュールを選択するだけです。CMK がローテーションの対象になると、新しいバッキングキーが作成され、情報を保護するための新しいすべての要求のアクティブなキーとしてマークされます。古いバッキングキーは、このキーを使用して暗号化された既存の暗号テキスト値を復号化するために使用可能なままです。CMK をより頻繁にローテーションするために、次のセクションで説明するように、`UpdateAlias` を呼び出して、エイリアスを新しい CMK にポイントすることもできます。`UpdateAlias` メソッドは、カスタマーマネージド型 CMK とインポートされたキーマテリアルを使用する CMK の両方で機能します。AWS は、キーのローテーションの頻度は法律、規制、企業のポリシーに大きく依存していることを発見しました。

キーのエイリアス

キーエイリアスを使用すると、基盤となるリージョン固有のキー ID とキー ARN からキーユーザーを抽出することができます。承認された個人は、アプリケーションがリージョンまたはローテーションスケジュールから独立して特定の CMK を使用できるようにするキーのエイリアスを作成できます。したがって、マルチリージョンのアプリケーションが、同じキーのエイリアスを使用して、キー ID またはキー ARN を気にすることなく、複数のリージョン内の KMS キーを参照することができます。また、特定のキーエイリアスを別の CMK にポイントすることで、CMK のローテーションを手動でトリガーすることもできます。ドメインネームサービス (DNS) が IP アドレスの抽象化を可能にするのと同様に、キーのエイリアスもキー ID に対して同じことを行います。キーのエイリアスを作成する場合、エイリアス/<環境>-<機能>-<サービスチーム>などのアカウント全体に適用できるような命名規則を決定することをお勧めします。

CMK エイリアスはポリシー内では使用できないことに注意してください。これは、エイリアスのキーへのマッピングをポリシー外で操作できるため、権限のエスカレーションが可能になるからです。したがって、KMS のキーポリシー、IAM ポリシー、KMS の付与では、キー ID を使用する必要があります。

大規模環境での AWS KMS の使用

前述のように、特定のクラスのデータに対して少なくとも 1 つの CMK を使用することがベストプラクティスです。これは、キーへの権限を制限するポリシー、つまり許可されたユーザーへのデータを定義するのに役立ちます。特定のデータ分類内でより強力なセキュリティ管理を提供するために、複数の CMK にわたってデータを配布することもできます。

AWS は、エンベロープ暗号化を使用して KMS の実装を拡張することを推奨しています。エンベロープ暗号化は、一意のデータキーで平文データを暗号化した後、キー暗号化キー (KEK) でデータキーを暗号化する方法です。AWS KMS 内では、CMK は KEK です。データキーでメッセージを暗号化し、CMK でデータキーを暗号化することができます。次に、暗号化されたデータキーを暗号化されたメッセージとともに保存することができます。繰り返し使用するために、データキーの平文バージョンをキャッシュして、AWS KMS へのリクエスト数を減らすことができます。さらに、エンベロープの暗号化は、災害対策のアプリケーションを設計するのに役立ちます。リージョン間で暗号化されたデータをそのまま移動し、リージョン固有の CMK でデータキーを再暗号化するだけで済みます。

AWS 暗号化チームは、効率的な方法で AWS KMS を使いやすくする AWS Encryption SDK をリリースしました。この SDK は、AWS KMS を使用するための低レベルの詳細を透過的に実装します。また、

使用後にデータキーを保護するための開発者オプションを提供し、機密データの暗号化がアプリケーションのパフォーマンスに大きな影響を与えないようにします。

データ保護

データ保護機能は、組織内で AWS KMS を使用して機密情報を保護するための AWS の一般的なユースケースの一部を扱います。

AWS KMS の一般的なユースケース

AWS KMS を使用する PCI データの暗号化

AWS KMS のセキュリティと品質管理は、PCI DSS レベル 1 認定の要件を満たすために検証、認定されているので、AWS KMS CMK でプライマリアカウント番号 (PAN) データを直接暗号化することができます。CMK を使用してデータを直接暗号化すると、暗号化ライブラリを管理する負担が軽減されます。さらに、AWS KMS から CMK をエクスポートすることができないので、暗号化キーが安全でない方法で格納される心配が緩和されます。すべての KMS リクエストが CloudTrail に記録されているので、CMK の使用は、CloudTrail ログを確認することで監査することができます。Payment Card Industry (PCI) データを保護するために直接 CMK を使用するアプリケーションを設計する場合は、1 秒あたりの要求数に注意することが重要です。

AWS KMS と Amazon S3 を使用する秘密管理

AWS KMS は主にキー管理機能を提供しますが、AWS KMS と Amazon S3 を活用して独自の秘密管理ソリューションを構築できます。

秘密を保持する新しい Amazon S3 バケットを作成します。バケットにバケットポリシーをデプロイして、許可された個人およびサービスだけにアクセスを制限します。バケットに保存される秘密は、秘密へのアクセスを細かくコントロールできるように、ファイルごとに事前に定義されたプレフィックスを使用します。それぞれの秘密は、S3 バケットに配置されると、特定のカスタマーマネージド型 KMS キーを使用して暗号化されます。さらに、このバケットに保存されている情報の機密性が高いため、S3 アクセスログまたは CloudTrail のデータイベントが監査目的で有効になります。次に、ユーザーまたはサービスが秘密にアクセスする必要がある場合、S3 バケット内のオブジェクトと KMS キーの両方を使用する権限を持つ AWS 内のアイデンティティを想定します。EC2 インスタンスで実行されるアプリケーションは、必要な権限を持つインスタンスのロールを使用します。

Lambda 環境変数の暗号化

デフォルトでは、環境変数を使用する Lambda 関数を作成または更新すると、それらの変数は AWS KMS を使用して暗号化されます。Lambda 関数を呼び出すと、それらの値は復号化され、Lambda コードで利用可能になります。Lambda にデフォルトの KMS キーを使用するか、選択した特定の CMK を指定することができます。

環境変数をさらに保護するには、「暗号化ヘルパーを有効にする」チェックボックスを選択する必要があります。このオプションを選択すると、環境変数もまた、選択した CMK を使用して個別に暗号化されます。次に、Lambda 関数は、必要な暗号化された環境変数を個別に復号化する必要があります。

Systems Manager の Parameter Store でのデータの暗号化

Amazon EC2 Systems Manager は、大規模な管理タスクを自動化するのに役立つ機能の集まりです。パスワード、ライセンスキー、証明書などの機密設定データを効率的に保存および参照するために、Parameter Store では、安全な文字列パラメータで機密情報を保護できます。

安全な文字列は、安全に保存および参照する必要のある機密データです。ドメイン結合パスワードやライセンスキーなど、ユーザーに平文で変更または参照させたくないデータがある場合は、その値を Secure String データ型を使用して指定します。以下のような状況では、安全な文字列を使用する必要があります。

- コマンド、関数、エージェントログ、CloudTrail ログでクリアテキストとして値を公開することなく、AWS サービス全体でデータ/パラメータを使用したい場合。
- 機密データへのアクセス権を持つユーザーをコントロールする必要がある場合。
- 機密データにアクセスされたときに、CloudTrail を使用して監査できるようにする必要がある場合。
- 重要なデータに対して AWS レベルの暗号化が必要で、アクセスを管理するために独自の暗号化キーを持ちたい場合。

パラメータを作成するときにこのオプションを選択すると、Systems Manager はその値をコマンドに渡すときに暗号化し、管理されたインスタンスで処理するときに復号化します。暗号化は AWS KMS によって処理され、Systems Manager のデフォルトの KMS キーにすることも、パラメータごとに特定の CMK を指定することもできます。

AWS サービスでの保存データ暗号化の実施

組織で、特定の分類に合致するすべてのデータの暗号化が必要になる場合があります。特定のサービスに応じて、予防的または発見的な統制によってデータ暗号化ポリシーを適用することができます。Amazon S3 のような一部のサービスでは、ポリシーによって暗号化されていないデータを保存できない場合があります。他のサービスでは、最も効率的なメカニズムは、ストレージリソースの作成を監視し、暗号化が適切に有効になっているかチェックすることです。暗号化されていないストレージが作成された場合は、ストレージリソースの削除から管理者への通知まで、さまざまな応答の可能性があります。

Amazon S3 での保存データ暗号化

Amazon S3 を使用すると、アップロードされているすべてのオブジェクトを確実に暗号化する S3 バケットポリシーをデプロイすることができます。ポリシーは次のようになります。

```
{
   "Version":"2012-10-17",
   "Id":"PutObjPolicy",
   "Statement":[{
         "Sid":"DenyUnEncryptedObjectUploads",
         "Effect":"Deny",
         "Principal":"*",
         "Action":"s3:PutObject",
         "Resource":"arn:aws:s3:::YourBucket/*",
         "Condition":{
            "StringNotEquals":{
               "s3:x-amz-server-side-encryption":"aws:kms"
            }
         }
      }
   ]
}
```

これにより、すでにバケット内にあるオブジェクトは暗号化されません。このポリシーは、オブジェクトが暗号化されていない限り、新しいオブジェクトをバケットに追加しようとする試みを拒否します。このポリシーを適用する前にすでにバケット内にあるオブジェクトは、最初にアップロードされた方法に基づいて暗号化されたままであるか、暗号化されないままになります。

Amazon EBS での保存データ暗号化

暗号化された EBS ブートボリュームを利用する Amazon マシンイメージ (AMI) を作成し、AMI を使用して EC2 インスタンスを起動することができます。保存されたデータは、EBS ボリュームと EC2 インスタンス間のデータ転送パスと同様に暗号化されます。データは、必要に応じてそのインスタンスのハイパーバイザで復号化され、メモリにのみ保存されます。この機能は、ブートボリュームまたはデータボリュームに格納されているかどうかに関わらず、EBS ボリュームに格納されているすべてのデータが暗号化されていることを確認できるようにすることで、セキュリティ、コンプライアンス、監査の作業を支援します。さらに、この機能は AWS KMS を使用するため、暗号化キーのすべての使用を追跡して監査することができます。

EBS ボリュームが常に暗号化されることを確認するには、2 つの方法があります。`CreateVolume` コンテキストの一部としての暗号化フラグが IAM ポリシーによって「true」に設定されていることを確認できます。フラグが「true」でなければ、IAM ポリシーは個人が EBS ボリュームを作成するのを防ぐことができます。もう 1 つの方法は、EBS ボリュームの作成を監視することです。新しい EBS ボリュームが作成されると、CloudTrail はイベントを記録します。EBS ボリュームが暗号化されているかどうか、どの KMS キーが暗号化に使用されたかを確認するために、CloudTrail イベントによって Lambda 関数をトリガーすることができます。

AWS Lambda 関数は、暗号化されていないボリュームの作成にいくつかの異なる方法で対応できます。この関数は、暗号化オプションで `CopyImage` API を呼び出して、EBS ボリュームの新しい暗号化バージョンを作成し、それをインスタンスにアタッチして古いバージョンを削除することができます。一部の顧客は、暗号化されていないボリュームを持つ EC2 インスタンスを自動的に削除することを選択します。他の顧客は、ほとんどのインバウンド接続を妨げるセキュリティグループを適用してインスタンスを自動的に隔離することを選択します。Amazon Simple Notification Service (SNS) トピックに投稿する Lambda 関数を記述するのも簡単です。このトピックでは、管理者に手動による調査と介入を行うよう警告します。ほとんどの実施の応答は、人の介入なしに、プログラムで達成することができますし、そうする必要があります。

Amazon RDS での保存データ暗号化

Amazon Relational Database Service (RDS) は、Amazon EBS 暗号化をベースに構築され、データベースボリュームに対して完全なディスク暗号化を提供します。Amazon RDS で暗号化されたデータベースインスタンスを作成すると、Amazon RDS はデータベースを保存するために暗号化された EBS ボリュ

ームを作成します。ボリューム上に保存されているデータ、データベーススナップショット、自動バックアップ、および読み取りレプリカは、データベースインスタンスの作成時に指定した KMS CMK ですべて暗号化されます。

Amazon EBS と同様に、AWS Lambda 関数を設定して、CloudTrail からの `CreateDBInstance` API コールを介して新しい RDS インスタンスの作成を監視することができます。`CreateDBInstance` イベントで、`KmsKeyId` パラメータが予想される CMK に設定されていることを確認します。

インシデント対応

インシデント対応機能は、AWS KMS に関連する可能性があるインシデントを修復する組織の能力に重点を置いています。

AWS KMS のセキュリティの自動化

CMK の監視中に、特定のアクションが検出された場合、CMK を無効にするか、ローカルのセキュリティポリシーで指示されているその他のインシデント対応アクションを実行するように AWS Lambda 関数を設定することができます。AWS 内の自動化ツールを活用することにより、人間の介入なしで潜在的な露出を数分で遮断することができます。

CMK の削除と無効化

CMK を削除することは可能ですが、組織に大きな影響を与えます。最初に使用する予定のないキーで CMK の状態を無効に設定するだけで十分か検討する必要があります。これにより、将来の CMK の使用がすべて防止されます。ただし、CMK はまだ使用可能であり、必要に応じて再び有効にすることができます。無効化されたキーは AWS KMS によって保存されます。したがって、引き続きストレージの費用が発生します。暗号化されたデータ管理に自信が持てるまで、キーを削除するのではなく、キーを無効にすることを強く検討する必要があります。

キーの削除は、非常に慎重に検討する必要があります。対応する CMK が削除された場合、データを復号化することはできません。さらに、いったん CMK を削除すると、それは永遠に消え去ってしまいます。最終的に削除されると、AWS には削除された CMK を回復する手段はありません。AWS の他の重要な操作と同様に、CMK の削除に MFA を必要とするポリシーを適用する必要があります。

誤って CMK を削除しないようにするため、KMS は CMK が実際に削除されるまでに最低 7 日間の待機期間を設けています。この待機時間を最大 30 日まで延長することができます。待機期間中、CMK は引き続き KMS の「削除保留中」状態で保存されます。暗号化または復号化の操作には使用できません。暗号化または復号化のために「削除保留中」状態にあるキーを使用しようとすると、CloudTrail にログが記録されます。CloudTrail ログでこれらのイベントの Amazon CloudWatch のアラームを設定できます。これにより、必要に応じて削除プロセスをキャンセルすることができます。待機期間が終了するまで、CMK を「削除保留中」状態から回復し、無効または有効のいずれかの状態に復元することができます。

最後に、インポートされたキーマテリアルで CMK を使用している場合は、インポートされたキーマテリアルを直ちに削除することができます。これは、CMK をいくつかの方法で直接削除するのとは異なります。`DeleteImportedKeyMaterial` アクションを実行すると、AWS KMS によってキーマテリアルが削除され、CMK キーの状態がインポート保留中に変更されます。キーマテリアルを削除すると、CMK はすぐに使用できなくなります。待機期間はありません。CMK の使用を再度有効にするには、同じキーマテリアルを再度インポートする必要があります。キーマテリアルを削除すると、すぐに CMK に影響が及びますが、AWS サービスで使用されているデータ暗号化キーはすぐには影響を受けません。

たとえば、インポートされたマテリアルを使用する CMK が、SSE-KMS[8] を使用して S3 バケットに配置されているオブジェクトを暗号化するために使用されたとします。オブジェクトを S3 バケットにアップロードする直前に、インポートされたマテリアルを CMK に配置します。オブジェクトがアップロードされたら、その CMK からキーマテリアルを削除することができます。オブジェクトは暗号化された状態で S3 バケットに存在し続けますが、同じキーマテリアルが CMK に再度インポートされるまで誰もアクセスできなくなります。このフローでは、明らかに CMK からキーマテリアルをインポートおよび削除するための正確な自動化が必要ですが、環境内で追加レベルのコントロールを提供することができます。

まとめ

AWS KMS は、暗号化キーを集中管理する完全に管理されたサービスを組織に提供します。他の AWS サービスとのネイティブな統合により、AWS KMS は、保存して処理するデータを簡単に暗号化できます。

AWS KMS を適切に設計し実装するには、暗号化キーが安全であり、アプリケーションや認証されたユーザーが使用できることを確認する必要があります。さらに、キーの使用状況に関連する詳細ログを監査者に表示することもできます。

寄稿者

本書の執筆に当たり、次の人物および組織が寄稿しました。

- Matthew Bretan、上級セキュリティコンサルタント、AWS プロフェッショナルサービス
- Sree Pisharody、シニア製品マネージャ -Technical、AWS Cryptography
- Ken Beer、シニアソフトウェア開発マネージャ、AWS Cryptography
- Brian Wagner、セキュリティコンサルタント、AWS プロフェッショナルサービス
- Eugene Yu、管理コンサルタント、AWS プロフェッショナルサービス
- Michael St.Onge、グローバルクラウドセキュリティアーキテクト、AWS プロフェッショナルサービス
- Balaji Palanisamy、シニアコンサルタント、AWS プロフェッショナルサービス
- Jonathan Rault、シニアコンサルタント、AWS プロフェッショナルサービス
- Reef Dsouza、コンサルタント、AWS プロフェッショナルサービス
- Paco Hope、プリンシパルコンサルタント、AWS プロフェッショナルサービス

文書の改訂

このホワイトペーパーの最新のバージョンは、下記をご覧ください。

https://d0.awsstatic.com/whitepapers/KMS-Best-Practices.pdf

注記

[1] http://docs.aws.amazon.com/kms/latest/developerguide/overview.html

[2] https://d0.awsstatic.com/whitepapers/KMS-Cryptographic-Details.pdf

[3] https://d0.awsstatic.com/whitepapers/aws_cloud_adoption_framework.pdf

[4] https://d0.awsstatic.com/whitepapers/AWS_CAF_Security_Perspective.pdf

[5] http://docs.aws.amazon.com/IAM/latest/UserGuide/introduction.html

[6] http://docs.aws.amazon.com/kms/latest/developerguide/key-policies.html#key-policy-default-allow-root-enable-iam

[7] http://docs.aws.amazon.com/kms/latest/developerguide/policy-conditions.html#conditions-kms-via-service

[8] http://docs.aws.amazon.com/kms/latest/developerguide/services-s3.html#sse