Enterprise Key Management

Implementation Guide

slack

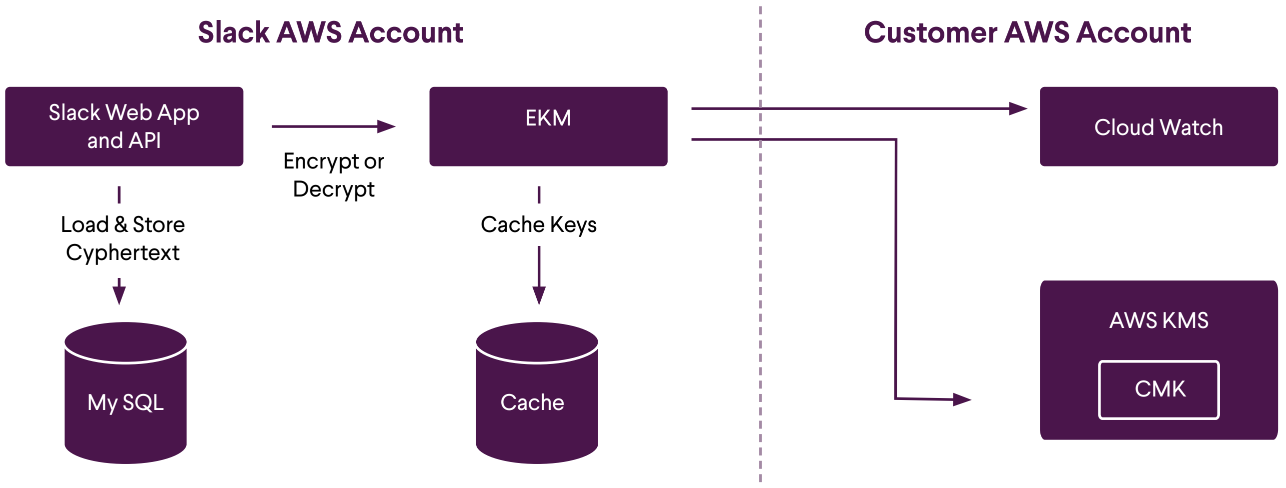
slack

Contents

1.0 Introduction

Welcome to Slack Enterprise Key Management (EKM). This document is designed to guide administrators, like yourself, through the enrollment, operation, and revocation phases of Slack EKM. Additionally, you will be designated a Slack resource to support you during the enrollment phase.

Slack EKM uses AWS Key Management Services (KMS) and AWS CloudWatch/CloudTrail Logs to allow you to retain control over your encryption keys. As such, this guide will walk you through the setup of Slack EKM, AWS KMS, and AWS CloudWatch/CloudTrail Logs.



For simplicity, we have broken up the guide into three primary phases:

Enrollment

Covers upfront configuration of your AWS account and the resources within it to support Slack EKM

Operation

Offers techniques for managing Slack EKM within your organization after initial enrollment

Revocation

Shares sample policy changes you may choose to invoke as your organization's risk posture evolves

2.0 Document Conventions

Below are key attributes that are referenced throughout the document; therefore, we want to define them upfront.

- `YOUR_AWS_ACCOUNT_NUMBER_DIGITS`: This placeholder must be replaced with your AWS account number, which is a 12-digit number that is sometimes written with dashes every four digits. When asked for this within the context of this guide, please omit the dashes.

- `YOUR_CMK_ARN`: This placeholder must be replaced with the ARN of your Customer Master Key (CMK). Once you've created your CMK, this can be found in the AWS console or in the response to the AWS KMS `ListKeys` API. It will be a string beginning with "`arn:aws:kms:`" and ending with a UUID.

- `ORGANIZATION_ID`: This placeholder must be replaced with your organization ID from the Slack API (or your Slack Technical Architect).

- `YOUR_ROLE_ARN`: This placeholder must be replaced with the ARN of an AWS Identity and Access Management (KMS) role you've configured for EKM logging. Once you've created the KMS role, this can be found in the AWS console or in the response to the AWS KMS `ListRoles` API.

- `YOUR_LOG_GROUP_NAME`: This placeholder must be replaced with the name of an AWS CloudWatch Logs log group. You do not need to create this ahead of time if you configure the role as recommended in the **Configure CloudWatch Logs** section below.

- `TEAM_ID`: This placeholder must be replaced with a team/workspace ID from the Slack API.

- `CHANNEL_ID` and `ANOTHER_CHANNEL_ID`: These placeholders must be replaced with a channel ID from the Slack API or desktop client URLs. See here for more info on how to collect channel IDs for disconnected channels.

- `FILE_ID`: This placeholder must be replaced with a file ID from the Slack API.

3.0 Enrollment

You should expect to follow this enrollment process twice: once for your sandbox Grid organization (for testing purposes) and once for your production Grid organization. Please note that members of your organization can continue using Slack throughout the enrollment process.

Step 1: Set up your AWS account

To set up your very own AWS account, follow the appropriate step below:

A If your company doesn't already have AWS accounts, simply sign up at https://aws.amazon.com/

B If your company already has a very evolved AWS infrastructure, we recommend Creating an AWS Account in Your Organization.

In any case, Slack strongly recommends configuring multifactor authentication for this and all AWS accounts.

Step 2: Create your Customer Master Key (CMK)

In order to create your own CMK, follow these steps:

1 Visit the encryption keys section of the AWS KMS console ensure that you select the region where you plan on having your Slack data hosted.

- USA : `us-east-1`
- France : `eu-west-3`
- Japan : `ap-northeast-1`
- Canada : `ca-central-1`
- Germany : `eu-central-1`
- UK: `eu-west-2`
- Australia: `ap-southeast-2`

2 Click **Create key** to begin a wizard that creates a key, master key, customer master key, CMK, or encryption key (frustratingly, AWS uses all of these terms interchangeably).

3 Select **Symmetric** for Key type

4 Enter an alias and a description:

a. For sandbox enrollments, accept the default Key Material Origin (under Advanced Options) to let AWS KMS generate the master key on your behalf.

b. For production enrollments, Slack strongly recommends importing external key material and storing a copy of the key material securely offline.

5 Accept the defaults on the Add Tags page (e.g., no need to add any tags unless you just want to add some tags) and the Define Key Administrative Permissions page (e.g., leave all the checkboxes unchecked).

6 On the Define Key Usage Permissions page under External Accounts, click **Add an External Account** and enter **152659312504** (Slack EKM AWS account number).

7 Click **Add an External Account** again and enter **429538831549** (Slack's AWS account number which serves files from S3).

8 Proceed and finish. The key policy that is generated on your behalf is sufficient for use with Slack EKM, but is over-permissive by granting Slack access to some AWS KMS APIs it doesn't use. See the **Step 3: Configure your key's policy** section below for a minimal key policy.

9 Click the alias you just created (in the list shown after finishing the wizard) to reveal the ARN. Note this for later use as `YOUR_CMK_ARN`.

For additional support, please reference:

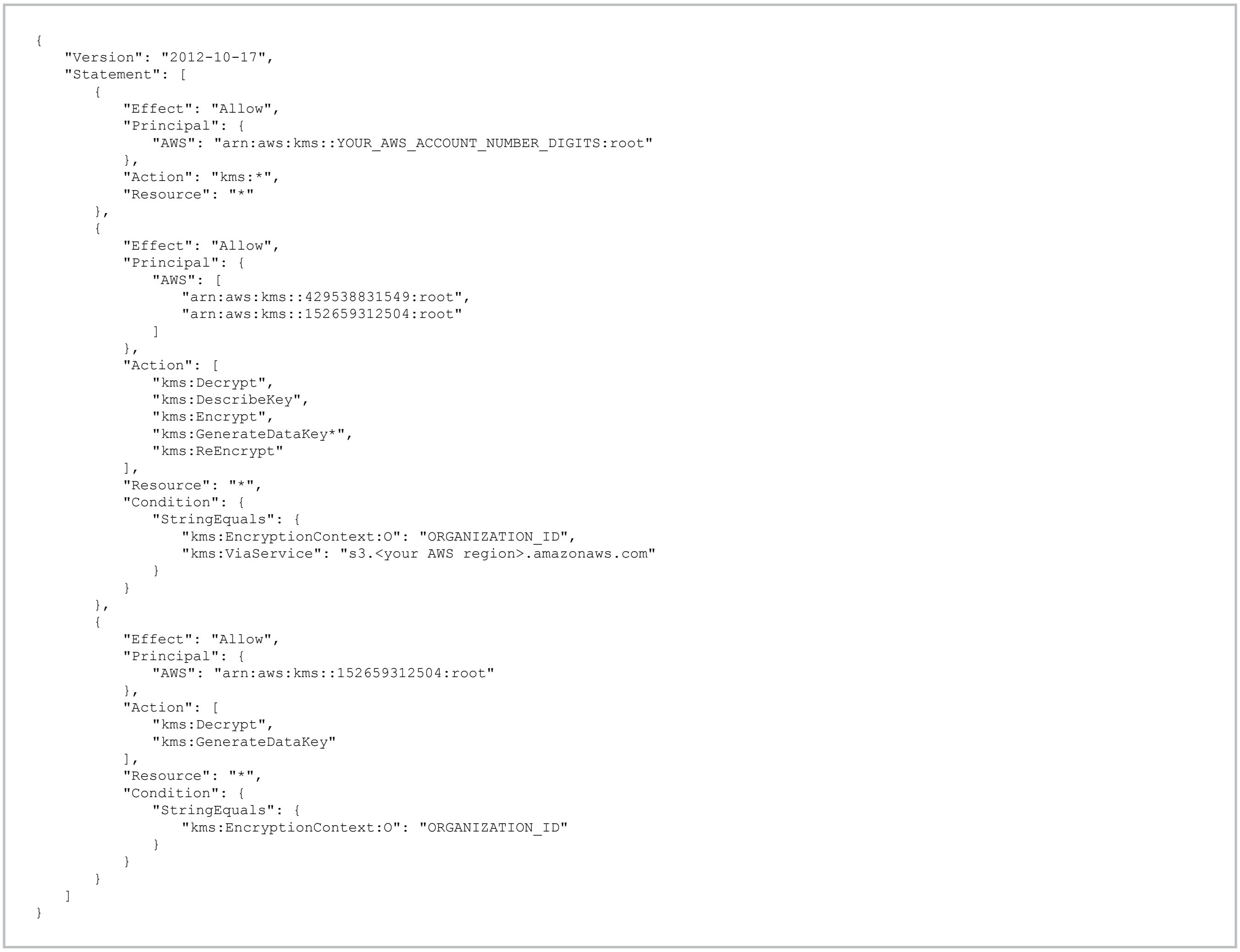
Creating Keys and Importing Key Material in AWS KMS

Step 3: Configure your key's policy

In order to configure your key's policy, go to the Customer Master Key page in the Encryption Keys section of the AWS KMS console. The first time you edit the wizard's generated key policy, you'll have to click **Switch to policy view** in order to see and edit the JSON representation.

It's important to note that this change and all future changes to your key policy may also be made via the AWS KMS `PutKeyPolicy` API or via infrastructure automation tools such as AWS CloudFormation/Terraform.

Below is the minimal key policy necessary for Slack EKM to function:



```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Effect": "Allow",
         "Principal": {
            "AWS": "arn:aws:kms::YOUR_AWS_ACCOUNT_NUMBER_DIGITS:root"
         },
         "Action": "kms:*",
         "Resource": "*"
      },
      {
         "Effect": "Allow",
         "Principal": {
            "AWS": [
               "arn:aws:kms::429538831549:root",
               "arn:aws:kms::152659312504:root"
            ]
         },
         "Action": [
            "kms:Decrypt",
            "kms:DescribeKey",
            "kms:Encrypt",
            "kms:GenerateDataKey*",
            "kms:ReEncrypt"
         ],
         "Resource": "*",
         "Condition": {
            "StringEquals": {
               "kms:EncryptionContext:O": "ORGANIZATION_ID",
               "kms:ViaService": "s3.<your AWS region>.amazonaws.com"
            }
         }
      },
      {
         "Effect": "Allow",
         "Principal": {
            "AWS": "arn:aws:kms::152659312504:root"
         },
         "Action": [
            "kms:Decrypt",
            "kms:GenerateDataKey"
         ],
         "Resource": "*",
         "Condition": {
            "StringEquals": {
               "kms:EncryptionContext:O": "ORGANIZATION_ID"
            }
         }
      }
   ]
}
```

In the key policy, you will need to include your Slack Organization ID, which should replace the "ORGANIZATION_ID" text in the above image. To get your Slack Organization ID, ask your Customer Success Manager, who can provide this information.

This key policy is the baseline from which all other revocation examples in this document are derived. Specifically, all the examples include the first statement, which allows you (as the owner of the CMK) to continue managing it, in addition to one or more statements that allow/deny Slack access to all or a subset of your organization's data keys.

For additional support, please reference:

Using Key Policies in AWS KMS

PutKeyPolicy - AWS Key Management Service

AWS::KMS::Key - AWS CloudFormation

AWS: aws_kms_key - Terraform by HashiCorp

Step 4: Provide your key family to Slack

Slack EKM enrollment is not self-service. Therefore, you need to provide `YOUR_CMK_ARN` to your Slack Solutions Engineer or Technical Architect to complete the key enrollment process.

- If you are setting up Slack EKM on a new Grid organization, all new data created in Slack (after key enrollment is complete) will be encrypted using your keys.
- If you are setting up Slack EKM on an existing Grid organization, all existing data created in Slack (before key enrollment is complete), and all new data created in Slack (after key enrollment is complete) will be encrypted using your keys.

Important note:

- If you are setting up Slack EKM on an existing Grid organization, the time to encrypt all your existing data using your keys will vary based on how much data you have.
- Migration of external workspaces to your Grid Org will be blocked if we are in the process of encrypting previously existing data.
- After the key enrollment is completed by your Slack Solution Engineer or Technical Architect, you can view your Grid organization's encryption status via the Key Management page in the Slack Org Dashboard.
- All Org Owners within your Grid organization will be notified via a Slackbot message once your Slack Solutions Engineer or Technical Architect completes the key enrollment process on your behalf.

Step 5: Configure logging for Slack EKM

Slack EKM supports two forms of logs via AWS CloudTrail and AWS CloudWatch.

Configure CloudTrail

The first form of Slack EKM logs are delivered via AWS CloudTrail, which logs every AWS API request, including the `GenerateDataKey` and `Decrypt` requests Slack makes to AWS KMS. Additionally, CloudTrail lLogs all S3 interactions related to file uploads with AWS KMS. It is important to note that Slack cannot tamper with this log as it is written atomically by AWS as API requests are serviced.

AWS CloudTrail makes a limited horizon of events available automatically. If you are part of an organization that wants to preserve this log for forensic purposes or wants to process it in your SIEM system, then visit the AWS CloudTrail console and click **Create trail.** The resulting form provides a lot of options for how the log is delivered and preserved for posterity.

- You must choose to log “All” or “Read-only” management events in order to capture AWS KMS requests
- You do not need to record any “Data events” and can skip that entire section
- When in doubt about “Storage location,” deliver the trail to Amazon S3, as that preserves your options for future processing

For additional support, please reference:

Logging AWS KMS API Calls with AWS CloudTrail

Configure CloudWatch Logs

The second form of Slack EKM logs are delivered via AWS CloudWatch Logs. These logs include key cache hits that never appear in AWS CloudTrail because they never prompt requests to AWS KMS. These logs also include the reason code for each encrypt or decrypt request. Below are the steps you need to take to configure your CloudWatch Logs:

1. Visit the roles section of the AWS KMS console and click **Create role.**
2. Choose “Another AWS account ” as the type of trusted entity.
3. Enter 152659312504 (Slack’s AWS account number) as the account ID.
4. Do not check the option to “Require external ID.”
5. Click **Next: Permissions.**
6. If this is your first time configuring Slack EKM, click **Create policy.** A new window will open. Select the JSON tab and paste the below policy in the text area revealed, choosing `YOUR_LOG_GROUP_NAME` as you see fit:

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Effect": "Allow",
         "Action": [
            "logs:CreateLogGroup",
            "logs:CreateLogStream",
            "logs:DescribeLogStreams",
            "logs:PutLogEvents"
         ],
         "Resource": [
            "arn:aws:logs:*:*:log-group:YOUR_LOG_GROUP_NAME",
            "arn:aws:logs:*:*:log-group:YOUR_LOG_GROUP_NAME:log-stream:*"
         ]
      }
   ]
}
```

7 Click **Review policy.**

8 On the next page, give your policy a name and click **Create policy.**

9 Close this window to return to role creation.

10 Click the refresh icon at the top right of the list of policies. Select your newly created policy and click **Next: Review.**

11 On the next page, give your role a name and click **Create role.**

12 Click the name of the role you just created (in the list shown after finishing the wizard) to reveal the ARN; note this for later use as `YOUR_ROLE_ARN`.

Step 6: Provide your logging configuration to Slack

Configuration of Slack EKM logging is not self-service. Therefore, you need to provide `YOUR_ROLE_ARN` and `YOUR_LOG_GROUP_NAME` to your Slack Solutions Engineer or Technical Architect to complete the logging enrollment process.

Important note:

- Configuring logging via AWS CloudWatch Logs will incur potentially significant charges in your AWS account. Please familiarize yourself with Amazon CloudWatch Pricing and arrange to monitor incurred costs before providing your logging configuration to Slack.
- All Org Owners within your Grid organization will be notified via a Slackbot message once your Slack Solutions Engineer or Technical Architect completes the logging enrollment process on your behalf.

For additional support, please reference:

Cost Explorer

Cost Explorer/Monthly costs by service

4.0 Operation

This section offers operational guidance to support you after enrolling in Slack EKM, with a focus on monitoring AWS logs and key rotation.

Monitor logs

The AWS CloudWatch Logs console provides rudimentary browsing capabilities; therefore, detailed analysis or integration into an existing SIEM system demands that logs be extracted from AWS CloudWatch Logs. There are several facilities you may use to achieve this, including:

1. Periodic exports to Amazon S3.
2. Real-time streaming to AWS Elasticsearch Service, AWS Kinesis, or AWS Lambda.

Important note: Logs may remain stored in AWS CloudWatch Logs forever or you may configure them to be automatically deleted once they reach a certain age.

For additional support, please reference:

What is Amazon CloudWatch Logs?

Working with Log Groups and Log Streams

Cost Explorer

Cost Explorer/Monthly costs by service

Create metrics from your logs

Monitoring Slack EKM request by request is a daunting prospect. That's why Slack recommends that organizations at least monitor the *volume* of activity as a crude but useful visualization of normal operation. AWS CloudWatch Logs by default emits metrics named `IncomingLogEvents` and `IncomingBytes` for each log group.

You can also decide to monitor specific reason codes, specific channels, or specific workspaces more closely as befits your organization's risk management program:

1. Start from the Log Groups section of the AWS CloudWatch console.
2. Select the radio button to the left of your log group.
3. Click **Create Metric Filter** to get started. The interactive filter testing on the next page will help you hone your filter to monitor exactly what you need.

For additional support, please reference:

Searching and Filtering Log Data

Example: Count Log Events

Example: Count Occurrences of a Term

Reconfigure CloudWatch Logs

It may become desirable at some time to reconfigure Slack EKM's use of AWS CloudWatch Logs. In cases in which you desire to configure AWS CloudWatch Logs for the first time, or to change the configured KMS role or log group name, revisit the instructions in the following steps, which are spelled out above:

1. Step 5: Configure Logging for Slack EKM > Configure CloudWatch Logs
2. Step 6: Provide your logging configuration to Slack

Disable CloudWatch Logs

It may likewise become desirable to stop Slack EKM writing to AWS CloudWatch Logs (most likely because your organization determined AWS CloudWatch Logs wasn't worth the cost for the marginal benefit over AWS CloudTrail). If so, engage your Slack Solutions Engineer or Technical Architect.

Rotate your Customer Master Key (CMK)

Due to a security incident or internal policies related to key hygiene, you may need to rotate your CMK. In order to do so, simply revisit the instructions in the following sections:

1. Step 2: Create your Customer Master Key (CMK)
2. Step 3: Configure your key's policy
3. Step 4: Provide your key family to Slack

Shortly after your Slack Solutions Engineer or Technical Architect has completed the key rotation process on your behalf, Slack's short-term key cache will fully expire and the new CMK will be used to encrypt all newly created data.

For any previously created data (before key rotation was completed), Slack will go back and rekey the data with your new CMK. You can view your Grid organization's encryption status via the Key Management page in the Slack Org Dashboard.

Important notes:

- As with the enrollment process, Slack recommends first testing the key rotation process in your sandbox Grid organization.
- Do not revoke Slack's access to the old CMK before the rekey process is complete as it will result in a loss of access to that data for your members.
- Members of your organization can continue using Slack throughout the key rotation process.
- All Org Owners within your Grid organization will be notified via a Slackbot message once your Slack Solutions Engineer or Technical Architect completes the key rotation process on your behalf.

For additional support, please reference:

Rotating Customer Master Keys

5.0 Revocation

Revoke Slack’s key access

Due to security or compliance risk, you may need to revoke Slack’s access to your organization’s keys. In order to reduce the disruption to your organization, we have provided the ability to respond to risks with targeted revocation of Slack’s key access.

Slack supports revoking key access at the following levels:

- ✓ Organization
- ✓ Workspace
- ✓ Channel
- ✓ Hour
- ✓ File

Important notes:

- Slack recommends testing revocation policies in your sandbox Grid organization in order to ensure it achieves the desired behavior without having an impact on end user experience.

- Additionally, every time you make a change to your AWS policy in order to revoke or restore Slack’s key access, we recommend that you force all clients to reload to ensure that the necessary changes are reflected in Slack. You can do this by clicking the Clear Cache button on the Key Management page in the Slack Org Dashboard.

- To gather `CHANNEL_ID` for disconnected channels, you can use: audit logs, DLP, and the admin.conversations.ekm.listOriginalConnectedChannelInfo method.

- Finally, users must be on the latest version of mobile and desktop clients in order to ensure proper revocation behavior (Slack recommends at least 3.61 for iOS and 2.76.1 for Android).

Revoke Slack's key access at the organization level

In order to revoke Slack's key access entirely, you must do so at the organization level.

1. Visit the encryption keys section of the AWS KMS console.
2. Find your customer master key.
3. Edit its policy to change the statement that allows Slack key access to instead deny key access, thus:

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Effect": "Allow",
         "Principal": {
            "AWS": "arn:aws:kms::YOUR_AWS_ACCOUNT_NUMBER_DIGITS:root"
         },
         "Action": "kms:*",
         "Resource": "*"
      },
      {
         "Effect": "Deny",
         "Principal": {
            "AWS": [
               "arn:aws:kms::429538831549:root",
               "arn:aws:kms::152659312504:root"
            ]
         },
         "Action": [
          "kms:Decrypt",
          "kms:DescribeKey",
          "kms:Encrypt",
          "kms:GenerateDataKey*",
          "kms:ReEncrypt"
         ],
         "Resource": "*",
         "Condition": {
            "StringEquals": {
               "kms:EncryptionContext:O": "ORGANIZATION_ID",
               "kms:ViaService": "s3.<your AWS region>.amazonaws.com"
            }
         }
      },
      {
         "Effect": "Deny",
         "Principal": {
            "AWS": "arn:aws:kms::152659312504:root"
         },
         "Action": [
            "kms:Decrypt",
            "kms:GenerateDataKey"
         ],
         "Resource": "*",
         "Condition": {
            "StringEquals": {
               "kms:EncryptionContext:O": "ORGANIZATION_ID"
            }
         }
      }
   ]
}
```

This policy is identical to the minimal key policy introduced before, except the change from `Allow` to `Deny` in the second and third statements.

Alternatively, you may remove the second and third statements entirely, which will also serve to deny Slack's key access entirely. These policies default to deny, so the absence of an allowing statement also results in denial. However, this is not recommended as it makes restoring Slack's key access later more difficult than changing `Deny` back to `Allow`.

Immediately after this change is made, Slack will be denied use of your CMK. Shortly after that, Slack's short-term key cache will fully expire and no messages or files will be decryptable. At this point, already-running Slack clients won't be able to load additional messages or files and new clients won't be able to load any messages or files at all. Also, Slack will proactively clear client-side caches upon detection of likely key access revocation.

Restore Slack's key access at the organization level

Ideally, after a risk is investigated and mitigated, key access can be restored. This is as simple as changing `Deny` to `Allow` in the key policy.

1. Visit the encryption keys section of the AWS KMS console.
2. Find your customer master key.
3. Edit its policy to change the statement that denies Slack key access to once again allow key access, like this:

Immediately after this change is made, Slack will be allowed to use your CMK. At this point, already-running Slack clients will once again be able to load additional messages and files and new clients will be able to load messages and files as usual.

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Effect": "Allow",
         "Principal": {
            "AWS": "arn:aws:kms::YOUR_AWS_ACCOUNT_NUMBER_DIGITS:root"
         },
         "Action": "kms:*",
         "Resource": "*"
      },
      {
         "Effect": "Allow",
         "Principal": {
            "AWS": [
               "arn:aws:kms::429538831549:root",
               "arn:aws:kms::152659312504:root"
            ]
         },
         "Action": [
            "kms:Decrypt",
            "kms:DescribeKey",
            "kms:Encrypt",
            "kms:GenerateDataKey*",
            "kms:ReEncrypt"
         ],
         "Resource": "*",
         "Condition": {
            "StringEquals": {
               "kms:EncryptionContext:0": "ORGANIZATION_ID",
               "kms:ViaService": "s3.<your AWS region>.amazonaws.com"
            }
         }
      },
      {
         "Effect": "Allow",
         "Principal": {
            "AWS": "arn:aws:kms::152659312504:root"
         },
         "Action": [
            "kms:Decrypt",
            "kms:GenerateDataKey"
         ],
         "Resource": "*",
         "Condition": {
            "StringEquals": {
               "kms:EncryptionContext:0": "ORGANIZATION_ID"
            }
         }
      }
   ]
}
```

More granular examples of revoking Slack's key access

You may add a large number of statements to your key policy to control Slack's use of your CMK in very fine-grained ways. There is no limit on the number of statements, but the JSON representation of the key policy must be less than 32 kilobytes in length.

The rest of this section lists example policies.

1. Visit the encryption keys section of the AWS KMS console.
2. Find your customer master key.
3. Edit its policy to incorporate any of the below statements. They can be combined to suit the needs of your organization.

Example 1: Revoke Slack’s key access at the workspace level

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Effect": "Allow",
         "Principal": {
            "AWS": "arn:aws:kms::YOUR_AWS_ACCOUNT_NUMBER_DIGITS:root"
         },
         "Action": "kms:*",
         "Resource": "*"
      },
      {
         "Effect": "Allow",
         "Principal": {
            "AWS": [
               "arn:aws:kms::429538831549:root",
               "arn:aws:kms::152659312504:root"
            ]
         },
         "Action": [
            "kms:Decrypt",
            "kms:DescribeKey",
            "kms:Encrypt",
            "kms:GenerateDataKey*",
            "kms:ReEncrypt"
         ],
         "Resource": "*",
         "Condition": {
            "StringEquals": {
               "kms:EncryptionContext:O": "ORGANIZATION_ID",
               "kms:ViaService": "s3.<your AWS region>.amazonaws.com"
            }
         }
      },
      {
         "Effect": "Allow",
         "Principal": {
            "AWS": "arn:aws:kms::152659312504:root"
         },
         "Action": [
            "kms:Decrypt",
            "kms:GenerateDataKey"
         ],
         "Resource": "*",
         "Condition": {
            "StringEquals": {
               "kms:EncryptionContext:O": "ORGANIZATION_ID"
            }
         }
      },
      {
         "Effect": "Deny",
         "Principal": {
            "AWS": "arn:aws:kms::152659312504:root"
         },
         "Action": [
            "kms:Decrypt",
            "kms:GenerateDataKey"
         ],
         "Resource": "*",
         "Condition": {
            "StringEquals": {
               "kms:EncryptionContext:T": "TEAM_ID"
            }
         }
      }
   ]
}
```

Example 2: Revoke Slack’s key access at the channel level *

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:kms::YOUR_AWS_ACCOUNT_NUMBER_DIGITS:root"
            },
            "Action": "kms:*",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": [
                    "arn:aws:kms::429538831549:root",
                    "arn:aws:kms::152659312504:root"
                ]
            },
            "Action": [
                "kms:Decrypt",
                "kms:DescribeKey",
                "kms:Encrypt",
                "kms:GenerateDataKey*",
                "kms:ReEncrypt"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "kms:EncryptionContext:O": "ORGANIZATION_ID",
                    "kms:ViaService": "s3.<your AWS region>.amazonaws.com"
                }
            }
        },
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:kms::152659312504:root"
            },
            "Action": [
                "kms:Decrypt",
                "kms:GenerateDataKey"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "kms:EncryptionContext:O": "ORGANIZATION_ID"
                }
            }
        },
        {
            "Effect": "Deny",
            "Principal": {
                "AWS": "arn:aws:kms::152659312504:root"
            },
            "Action": [
                "kms:Decrypt",
                "kms:GenerateDataKey"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "kms:EncryptionContext:C": "CHANNEL_ID"
                }
            }
        }
    ]
}
```

*See here for more info on how to collect `CHANNEL_ID` for disconnected channels

Example 3: Revoke Slack's key access to two channels

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:kms::YOUR_AWS_ACCOUNT_NUMBER_DIGITS:root"
            },
            "Action": "kms:*",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": [
                    "arn:aws:kms::429538831549:root",
                    "arn:aws:kms::152659312504:root"
                ]
            },
            "Action": [
                "kms:Decrypt",
                "kms:DescribeKey",
                "kms:Encrypt",
                "kms:GenerateDataKey*",
                "kms:ReEncrypt"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "kms:EncryptionContext:O": "ORGANIZATION_ID",
                    "kms:ViaService": "s3.<your AWS region>.amazonaws.com"
                }
            }
        },
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:kms::152659312504:root"
            },
            "Action": [
                "kms:Decrypt",
                "kms:GenerateDataKey"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "kms:EncryptionContext:O": "ORGANIZATION_ID"
                }
            }
        },
        {
            "Effect": "Deny",
            "Principal": {
                "AWS": "arn:aws:kms::152659312504:root"
            },
            "Action": [
                "kms:Decrypt",
                "kms:GenerateDataKey"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "kms:EncryptionContext:C": [
                        "CHANNEL_ID",
                        "ANOTHER_CHANNEL_ID"
                    ]
                }
            }
        }
    ]
}
```

Example 4: Revoke Slack’s key access to uploaded S3 files

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Effect": "Allow",
         "Principal": {
            "AWS": "arn:aws:kms::YOUR_AWS_ACCOUNT_NUMBER_DIGITS:root"
         },
         "Action": "kms:*",
         "Resource": "*"
      },
      {
         "Effect": "Deny",
         "Principal": {
            "AWS": [
               "arn:aws:kms::429538831549:root",
               "arn:aws:kms::152659312504:root"
            ]
         },
         "Action": [
            "kms:Decrypt",
            "kms:DescribeKey",
            "kms:Encrypt",
            "kms:GenerateDataKey*",
            "kms:ReEncrypt"
         ],
         "Resource": "*",
         "Condition": {
            "StringEquals": {
               "kms:EncryptionContext:O": "ORGANIZATION_ID",
               "kms:ViaService": "s3.<your AWS region>.amazonaws.com"
            }
         }
      },
      {
         "Effect": "Allow",
         "Principal": {
            "AWS": "arn:aws:kms::152659312504:root"
         },
         "Action": [
            "kms:Decrypt",
            "kms:GenerateDataKey"
         ],
         "Resource": "*",
         "Condition": {
            "StringEquals": {
               "kms:EncryptionContext:O": "ORGANIZATION_ID"
            }
         }
      }
   ]
}
```

Example 5: Revoke Slack’s key access to a specific hour

Important note: Time should be expressed as a four-digit year, two-digit month, and two-digit day, followed by the uppercase letter ‘T’ and a two-digit hour from a 24-hour UTC clock. *(This is a prefix of an ISO 8601 timestamp.)*

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Effect": "Allow",
         "Principal": {
            "AWS": "arn:aws:kms::YOUR_AWS_ACCOUNT_NUMBER_DIGITS:root"
         },
         "Action": "kms:*",
         "Resource": "*"
      },
      {
         "Effect": "Allow",
         "Principal": {
            "AWS": [
               "arn:aws:kms::429538831549:root",
               "arn:aws:kms::152659312504:root"
            ]
         },
         "Action": [
            "kms:Decrypt",
            "kms:DescribeKey",
            "kms:Encrypt",
            "kms:GenerateDataKey*",
            "kms:ReEncrypt"
         ],
         "Resource": "*",
         "Condition": {
            "StringEquals": {
               "kms:EncryptionContext:O": "ORGANIZATION_ID",
               "kms:ViaService": "s3.<your AWS region>.amazonaws.com"
            }
         }
      },
      {
         "Effect": "Allow",
         "Principal": {
            "AWS": "arn:aws:kms::152659312504:root"
         },
         "Action": [
            "kms:Decrypt",
            "kms:GenerateDataKey"
         ],
         "Resource": "*",
         "Condition": {
            "StringEquals": {
               "kms:EncryptionContext:O": "ORGANIZATION_ID"
            }
         }
      },
      {
         "Effect": "Deny",
         "Principal": {
            "AWS": "arn:aws:kms::152659312504:root"
         },
         "Action": [
            "kms:Decrypt",
            "kms:GenerateDataKey"
         ],
         "Resource": "*",
         "Condition": {
            "StringEquals": {
               "kms:EncryptionContext:H": "2018-07-20T14"
            }
         }
      }
   ]
}
```

Example 6: Revoke Slack’s key access to a whole month

Important note: Implement this by taking advantage of the wildcard support in the `StringLike` condition operator.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:kms::YOUR_AWS_ACCOUNT_NUMBER_DIGITS:root"
            },
            "Action": "kms:*",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": [
                    "arn:aws:kms::429538831549:root",
                    "arn:aws:kms::152659312504:root"
                ]
            },
            "Action": [
                "kms:Decrypt",
                "kms:DescribeKey",
                "kms:Encrypt",
                "kms:GenerateDataKey*",
                "kms:ReEncrypt"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "kms:EncryptionContext:O": "ORGANIZATION_ID",
                    "kms:ViaService": "s3.<your AWS region>.amazonaws.com"
                }
            }
        },
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:kms::152659312504:root"
            },
            "Action": [
                "kms:Decrypt",
                "kms:GenerateDataKey"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "kms:EncryptionContext:O": "ORGANIZATION_ID"
                }
            }
        },
        {
            "Effect": "Deny",
            "Principal": {
                "AWS": "arn:aws:kms::152659312504:root"
            },
            "Action": [
                "kms:Decrypt",
                "kms:GenerateDataKey"
            ],
            "Resource": "*",
            "Condition": {
                "StringLike": {
                    "kms:EncryptionContext:H": "2018-07-*"
                }
            }
        }
    ]
}
```

Example 7: Revoke Slack’s key access to two channels, but only for two months

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:kms::YOUR_AWS_ACCOUNT_NUMBER_DIGITS:root"
            },
            "Action": "kms:*",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": [
                    "arn:aws:kms::429538831549:root",
                    "arn:aws:kms::152659312504:root"
                ]
            },
            "Action": [
                "kms:Decrypt",
                "kms:DescribeKey",
                "kms:Encrypt",
                "kms:GenerateDataKey*",
                "kms:ReEncrypt"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "kms:EncryptionContext:O": "ORGANIZATION_ID",
                    "kms:ViaService": "s3.<your AWS region>.amazonaws.com"
                }
            }
        },
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:kms::152659312504:root"
            },
            "Action": [
                "kms:Decrypt",
                "kms:GenerateDataKey"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "kms:EncryptionContext:O": "ORGANIZATION_ID"
                }
            }
        },
        {
            "Effect": "Deny",
            "Principal": {
                "AWS": "arn:aws:kms::152659312504:root"
            },
            "Action": [
                "kms:Decrypt",
                "kms:GenerateDataKey"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "kms:EncryptionContext:C": [
                        "CHANNEL_ID",
                        "ANOTHER_CHANNEL_ID"
                    ]
                },
                "StringLike": {
                    "kms:EncryptionContext:H": [
                        "2018-06-*",
                        "2018-07-*"
                    ]
                }
            }
        }
    ]
}
```

Example 8: Revoke Slack's key access to a single file

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:kms::YOUR_AWS_ACCOUNT_NUMBER_DIGITS:root"
            },
            "Action": "kms:*",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": [
                    "arn:aws:kms::429538831549:root",
                    "arn:aws:kms::152659312504:root"
                ]
            },
            "Action": [
                "kms:Decrypt",
                "kms:DescribeKey",
                "kms:Encrypt",
                "kms:GenerateDataKey*",
                "kms:ReEncrypt"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "kms:EncryptionContext:O": "ORGANIZATION_ID",
                    "kms:ViaService": "s3.<your AWS region>.amazonaws.com"
                }
            }
        },
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:kms::152659312504:root"
            },
            "Action": [
                "kms:Decrypt",
                "kms:GenerateDataKey"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "kms:EncryptionContext:O": "ORGANIZATION_ID"
                }
            }
        },
        {
            "Effect": "Deny",
            "Principal": {
                "AWS": "arn:aws:kms::429538831549:root"
            },
            "Action": "kms:*",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "kms:EncryptionContext:F": "FILE_ID",
                    "kms:ViaService": "s3.<your AWS region>.amazonaws.com"
                }
            }
        }
    ]
}
```

For additional support, please reference:

Using Policy Conditions with AWS KMS

KMS JSON Policy Elements: Condition

KMS JSON Policy Elements: String Condition Operators

Creating a Condition That Tests Multiple Key Values (Set Operations)

General guidance on restoring Slack's key access

Ideally, after a risk is investigated and mitigated, key access can be restored. This is as simple as changing `Deny` to `Allow` in the key policy.

1. Visit the encryption keys section of the AWS KMS console.

2. Find your customer master key.

3. Edit its policy to change the statement that denies Slack key access to once again allow key access, like this:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:kms::YOUR_AWS_ACCOUNT_NUMBER_DIGITS:root"
            },
            "Action": "kms:*",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": [
                    "arn:aws:kms::429538831549:root",
                    "arn:aws:kms::152659312504:root"
                ]
            },
            "Action": [
                "kms:Decrypt",
                "kms:DescribeKey",
                "kms:Encrypt",
                "kms:GenerateDataKey*",
                "kms:ReEncrypt"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "kms:EncryptionContext:O": "ORGANIZATION_ID",
                    "kms:ViaService": "s3.<your AWS region>.amazonaws.com"
                }
            }
        },
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:kms::152659312504:root"
            },
            "Action": [
                "kms:Decrypt",
                "kms:GenerateDataKey"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "kms:EncryptionContext:O": "ORGANIZATION_ID"
                }
            }
        }
    ]
}
```

6.0 Additional Support

For additional support, please check out:

The various '**For additional support, please reference**' subsections above

AWS KMS Developer Guide

AWS Key Management Service Cryptographic Details