

Always Encrypted related to AKV rotate key permission

Last updated by | Vitor Tomaz | Jun 8, 2022 at 5:22 AM PDT

Contents

- [Issue](#)
- [Investigation/Analysis](#)
- [Mitigation](#)
- [RCA Template \(optional\)](#)
- [Public Doc Reference \(optional\)](#)
- [Internal Reference \(optional\)](#)
- [Root Cause Classification](#)

Issue

Always Encrypted v1 scenarios where the SSMS v18 window crashes while trying to create CMK using AKV which has the Rotate key permission set

Investigation/Analysis

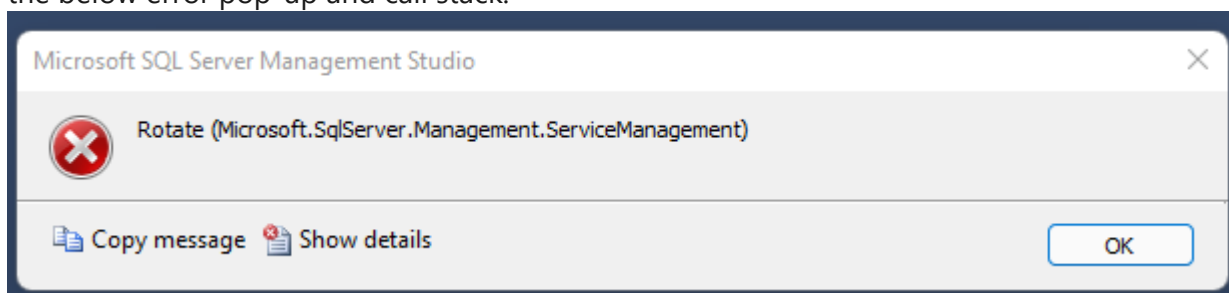
1. Configure the access policies on the AKV in the portal such that any of the below rotate key permissions

Rotation Policy Operations

- ☐ Select all
- ☒ Rotate
- ☐ Get Rotation Policy
- ☐ Set Rotation Policy

are set.

2. Connect to the server and database(general purpose) using SSMS v18. No need to have always encrypted enabled in the connection dialog box.
3. Try to create a CMK using the above AKV, which has any of the rotate key permissions in the access policies. a.On selecting the concerned AKV name after authentication, the SSMS CMK window crashes with the below error pop-up and call stack.



Message component:

- All messages
 - Rotate
 - Message
 - Program Location

Details:

Rotate (Microsoft.SqlServer.Management.ServiceManagement)

Program Location:

```

at Microsoft.SqlServer.Management.ServiceManagement.ResourceManagement.AzureKeyVaultKeyPermissionEnumConverter.ReadJson(JsonReader reader, Type objectType, Object existingValue, JsonSerializer serializer)
at Newtonsoft.Json.Serialization.JsonSerializerInternalReader.DeserializeConvertable(JsonConverter converter, JsonReader reader, Type objectType, Object existingValue)
at Newtonsoft.Json.Serialization.JsonSerializerInternalReader.PopulateList(IList list, JsonReader reader, JsonArrayContract contract, JsonProperty containerProperty, String id)
at Newtonsoft.Json.Serialization.JsonSerializerInternalReader.CreateList(JsonReader reader, Type objectType, JsonContract contract, JsonProperty member, Object existingValue, String id)
at Newtonsoft.Json.Serialization.JsonSerializerInternalReader.CreateValueInternal(JsonReader reader, Type objectType, JsonContract contract, JsonProperty member, JsonContainerContract containerContract, JsonProperty containerMember, Object existingValue)
at Newtonsoft.Json.Serialization.JsonSerializerInternalReader.SetPropertyValue(JsonProperty property, JsonConverter propertyConverter, JsonContainerContract containerContract, JsonProperty containerProperty, JsonReader reader, Object target)
at Newtonsoft.Json.Serialization.JsonSerializerInternalReader.PopulateObject(Object newObject, JsonReader reader, JsonObjectContract contract, JsonProperty member, String id)
at Newtonsoft.Json.Serialization.JsonSerializerInternalReader.CreateObject(JsonReader reader, Type objectType, JsonContract contract, JsonProperty member, JsonContainerContract containerContract, JsonProperty containerMember, Object existingValue)
at Newtonsoft.Json.Serialization.JsonSerializerInternalReader.CreateValueInternal(JsonReader reader, Type objectType, JsonContract contract, JsonProperty member, JsonContainerContract containerContract, JsonProperty containerMember, Object existingValue)
at Newtonsoft.Json.Serialization.JsonSerializerInternalReader.SetPropertyValue(JsonProperty property, JsonConverter propertyConverter, JsonContainerContract containerContract, JsonProperty containerProperty, JsonReader reader, Object target)
at Newtonsoft.Json.Serialization.JsonSerializerInternalReader.PopulateObject(Object newObject, JsonReader reader, JsonObjectContract contract, JsonProperty member, String id)
at Newtonsoft.Json.Serialization.JsonSerializerInternalReader.CreateObject(JsonReader reader, Type objectType, JsonContract contract, JsonProperty member, JsonContainerContract containerContract, JsonProperty containerMember, Object existingValue)
at Newtonsoft.Json.Serialization.JsonSerializerInternalReader.CreateValueInternal(JsonReader reader, Type objectType, JsonContract contract, JsonProperty member, JsonContainerContract containerContract, JsonProperty containerMember, Object existingValue)
at Newtonsoft.Json.Serialization.JsonSerializerInternalReader.PopulateList(IList list, JsonReader reader, JsonArrayContract contract, JsonProperty containerProperty, String id)
at Newtonsoft.Json.Serialization.JsonSerializerInternalReader.CreateList(JsonReader reader, Type objectType, JsonContract contract, JsonProperty member, Object existingValue, String id)

```

Mitigation

In the meantime, we recommend customers not have any of these 3 rotate permissions(Rotate, Get Rotation Policy, Set Rotation Policy) set in the access policy of the AKV they use for the Always Encrypted operations. This should help prevent any more CRIs related to this issue.

RCA Template (optional)


TBD

Public Doc Reference (optional)

TBD

Internal Reference (optional)

Master ICM to link support cases for reference [ICM](#) 

We have a bug in the AE scrub board to track this work item SQL Security AlwaysEncrypted Backlog items [Backlog](#).  I have identified the fix for it and have a code review out, but customers might still want to use the current SSMS version and it might take time to get a new version out.

Root Cause Classification

Azure SQL DB/Security/Always Encrypted/Product Bug

How good have you found this content?



-