3. Implement Azure security

└ **3.2 Implement secure Azure solutions**

└ **3.2.2 Develop code that uses keys, secrets, and certificates stored in Azure Key Vault**

1. How do you retrieve a secret from Azure Key Vault using the Azure SDK?
2. How do you use a certificate from Key Vault in an HTTPS client or service?
3. How do you use Key Vault to perform cryptographic operations with stored keys?
4. What roles or permissions are needed to access keys, secrets, or certificates?
5. How do you handle secret rotation using Azure Key Vault?
6. What are the differences between software-protected and HSM-protected keys?
7. How do you access a certificate's private key from Azure Key Vault?
8. How do you manage access to Key Vault from an Azure Function or Web App?
9. What are best practices for using Key Vault in application code?
10. How do you configure Key Vault references in an ARM or Bicep deployment?

---

1. How do you retrieve a secret from Azure Key Vault using the Azure SDK?

```
var client = new SecretClient(new Uri(kvUrl), new DefaultAzureCredential());
KeyVaultSecret secret = await client.GetSecretAsync("MySecret");
string value = secret.Value;
```

Requires Key Vault Secrets User role and managed identity or credential.

---

2. How do you use a certificate from Key Vault in an HTTPS client or service?

Download the certificate as a PFX with private key:

```
var certClient = new CertificateClient(new Uri(kvUrl), new DefaultAzureCredential());
KeyVaultCertificateWithPolicy cert = await certClient.GetCertificateAsync("MyCert");
var x509 = new X509Certificate2(cert.Cer);
```

For private key use, export from a secret or use GetSecretAsync with content type application/x-pkcs12.

---

3. How do you use Key Vault to perform cryptographic operations with stored keys?

Use CryptographyClient:

```
var cryptoClient = new CryptographyClient(new Uri(keyId), new DefaultAzureCredential());
EncryptResult result = await cryptoClient.EncryptAsync(EncryptionAlgorithm.RsaOaep, data);
```

Key must allow crypto operations (e.g., encrypt, sign).

---

4. What roles or permissions are needed to access keys, secrets, or certificates?

- Secrets: Key Vault Secrets User
- Keys: Key Vault Crypto Service Encryption User, Key Vault Key User
- Certificates: Key Vault Certificates Officer
  Use RBAC or Key Vault access policies (legacy).

---

5. How do you handle secret rotation using Azure Key Vault?

- For manual rotation: update secret value and update app references.
- For automatic rotation (certs): configure lifetimeAction in certificate policy.
- Enable soft-delete and purge protection for rollback and audit.

6. What are the differences between software-protected and HSM-protected keys?

- *Software-protected*: Stored and processed in software; suitable for general use.
- *HSM-protected*: Backed by FIPS 140-2 Level 2+ compliant Hardware Security Modules; use for high-security needs like compliance-bound apps.

---

7. How do you access a certificate's private key from Azure Key Vault?

Download as a secret in PFX format:

```
var secret = await secretClient.GetSecretAsync("MyCert");
var certBytes = Convert.FromBase64String(secret.Value);
var cert = new X509Certificate2(certBytes, (string)null, X509KeyStorageFlags.Exportable);
```

Ensure certificate is imported with the private key.

---

8. How do you manage access to Key Vault from an Azure Function or Web App?

- Enable system-assigned identity
- Assign appropriate RBAC role (e.g., Key Vault Secrets User)
- Use DefaultAzureCredential in app code for auth
  No secrets stored in config needed.

---

9. What are best practices for using Key Vault in application code?

- Use DefaultAzureCredential
- Use caching to minimize latency and throttling
- Do not log secret values
- Handle retries and transient failures with SDK policies

---

10. How do you configure Key Vault references in an ARM or Bicep deployment?

Use @Microsoft.KeyVault reference in resource parameters:

```
"mySecret": {
  "reference": {
    "keyVault": {
      "id": "[resourceId('Microsoft.KeyVault/vaults', 'my-kv')]"
    },
    "secretName": "my-secret"
  }
}
```

Used to inject secrets at deploy time into app settings or parameters.