Cryptography in Cloud Security

Introduction

Cryptography is a fundamental component of cloud security, ensuring data confidentiality, integrity, and authenticity. This document covers how cloud providers use encryption, particularly focusing on Azure Key Vault, Transport Layer Security (TLS) & SSL Certificates, and provides step-by-step guides for generating and installing an SSL certificate on a local web server and enabling encryption for Azure Blob Storage.

How Cloud Providers Use Encryption (Azure Key Vault)

1. Overview of Azure Key Vault

Azure Key Vault is a cloud service provided by Microsoft that helps in securely storing and managing cryptographic keys, certificates, and secrets. It enhances data protection and compliance by using:

- Hardware Security Modules (HSMs) for key management.
- Access Policies for granular control.
- Integration with Azure services for automatic key and certificate management.

2. Key Features of Azure Key Vault

- Secret Management: Store API keys, passwords, certificates securely.
- Key Management: Generate and control encryption keys for data protection.
- Certificate Management: Securely manage SSL/TLS certificates.
- Access Control: Assign specific permissions to users and services.

3. Enabling Encryption with Azure Key Vault

- Step 1: Create an Azure Key Vault

```
az keyvault create --name <vault-name> --resource-group <resource-group> --location <location>
```

- Step 2: Store a Secret in Azure Key Vault

```
az keyvault secret set --vault-name <vault-name> --name <secret-name> --value <secret-value>
```

- Step 3: Retrieve a Secret from Azure Key Vault

```
az keyvault secret show --vault-name <vault-name> --name <secret-name>
```

Transport Layer Security (TLS) & SSL Certificates

1. What is TLS and SSL?

TLS (Transport Layer Security) and SSL (Secure Sockets Layer) are cryptographic protocols that provide secure communication over networks. TLS is the modern and more secure successor to SSL.

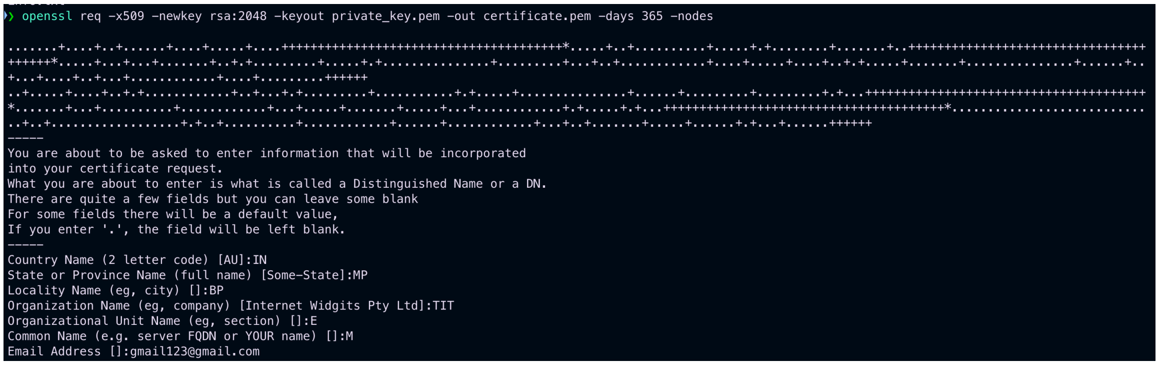
2. How SSL/TLS Works

1. A client (browser) requests a secure connection (HTTPS).
2. The server responds with an SSL/TLS certificate.
3. The client verifies the certificate's validity.
4. A secure encrypted connection is established using symmetric encryption.

Task 1: Generate a Self-Signed SSL Certificate and Install It on a Local Web Server

Step 1: Generate an SSL Certificate Using OpenSSL

openssl req -x509 -newkey rsa:2048 -keyout private_key.pem -out certificate.pem -days 365 -nodes



```
❯ openssl req -x509 -newkey rsa:2048 -keyout private_key.pem -out certificate.pem -days 365 -nodes

.......+....+..+......+....+.....+....+++++++++++++++++++++++++++++++++++++++*.....+..+..........+.....+.+........+.......+..+++++++++++++++++++++++++++++++++
++++++*.....+...+...+.......+..+.+.........+.....+.+...............+.........+...+..+...........+....+.....+....+..+.+.....+.......+...............+......+..
+...+....+..+...+...........+....+.........++++++
..+.....+....+..+.+...........+..+...+.+.........+...........+.+.....+...............+......+.........+.........+.+...+++++++++++++++++++++++++++++++++++++++
*.......+...+..........+............+...+.....+.......+.....+...+............+.+.....+.+...+++++++++++++++++++++++++++++++++++++++*.........................
..+..+.................+.+..+..........+............+......+...........+...+..+.......+.....+......+.+...+......++++++
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:IN
State or Province Name (full name) [Some-State]:MP
Locality Name (eg, city) []:BP
Organization Name (eg, company) [Internet Widgits Pty Ltd]:TIT
Organizational Unit Name (eg, section) []:E
Common Name (e.g. server FQDN or YOUR name) []:M
Email Address []:gmail123@gmail.com
```

Step 2: Install the SSL Certificate on a Web Server

For Apache:

- 1. Move the certificate and key files:

sudo mv certificate.pem /etc/ssl/certs/
sudo mv private_key.pem /etc/ssl/private/

- 2. Edit the Apache SSL configuration file:

```
sudo nano /etc/apache2/sites-available/default-ssl.conf
```

- 3. Add the following lines to enable SSL:

```
SSLEngine on
SSLCertificateFile /etc/ssl/certs/certificate.pem
SSLCertificateKeyFile /etc/ssl/private/private_key.pem
```

- 4. Restart Apache:

```
sudo a2enmod ssl
sudo systemctl restart apache2
```

Task 2: Enable Encryption for Azure Blob Storage

- Step 1: Enable Server-Side Encryption (SSE) in Azure Portal

1. Go to Azure Portal.
2. Navigate to Storage Accounts and select your account.
3. Under 'Security + Networking,' select 'Encryption.'
4. Ensure 'Microsoft-managed keys' are enabled (default).
5. Click 'Save.'

- Step 2: Enable Encryption Using Azure CLI

```
az storage account update --name <storage-account> --resource-group <resource-group> --encryption-services blob
```

- Step 3: Upload a File to an Encrypted Blob Container

```
az storage blob upload --account-name <storage-account> --container-name <container-name> --name file.txt --file file.txt --auth-mode key
```

- Step 4: Verify Encryption Status

```
az storage account show --name <storage-account> --query encryption
```

Conclusion

This document has explained cryptography in cloud security, covering Azure Key Vault, SSL/TLS certificates, and step-by-step implementation of self-signed SSL certificates for web servers and enabling encryption in Azure Blob Storage.