Asymmetric Key Encryption

Keith Bogart

CCIE #4923

- kbogart@ine.com
- @keithbogart1
- linkedin.com/in/keith-bogart-2a75042

CCIE Routing & Switching

Topic Overview

- Encryption Keys: Symmetric and Asymmetic
- Difference Between Public And Private Keys
- How Asymmetric Keys Are Used In Encryption
- Viewing a Public Key

Encryption Keys

- Generally speaking, there are two types of keys used for Cryptography;
  - Symmetric Keys
  - Asymmetric Keys
- Symmetric Keys: Same key is used for encrypting and decrypting the data
  - Like a shared password
  - Computationally inexpensive and fast
- Asymmetric Keys:
  - One key used for encrypting data, a different key used for decrypting data
  - Harder to compromise (more secure)
  - More CPU intensive to implement
  - Frequently referred to as an “RSA Keypair” or “Diffie-Hellman Keypair”

RSA isn’t the ONLY method available to create asymmetric encryption keys…just one of the more common ones.

Asymmetric Keys

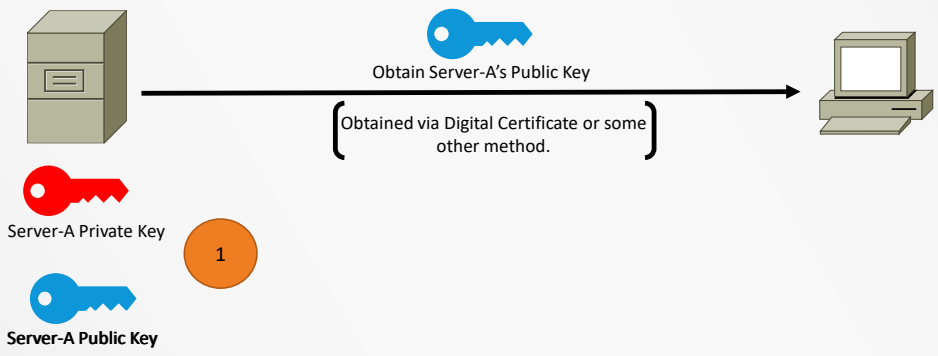
- Public Key
  - Freely accessible to the Public
  - Included in Digital Certificates

- Private Key
  - Highly Secure
  - Kept hidden…don't give it to anyone.

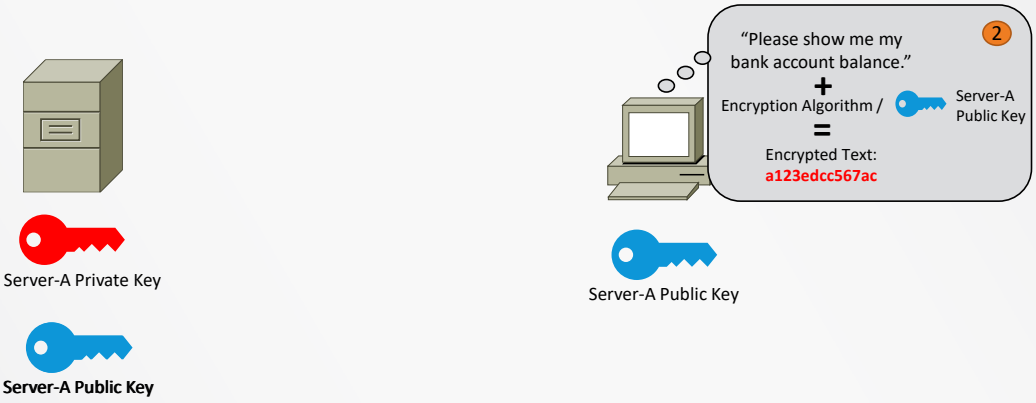
- Both keys mathematically related to each other (VERY complex)
- Data encrypted with one key can only be decrypted with the OTHER key.

Asymmetric Keys

Step-1: Server derives Public/Private Keypair and distributes Public Key

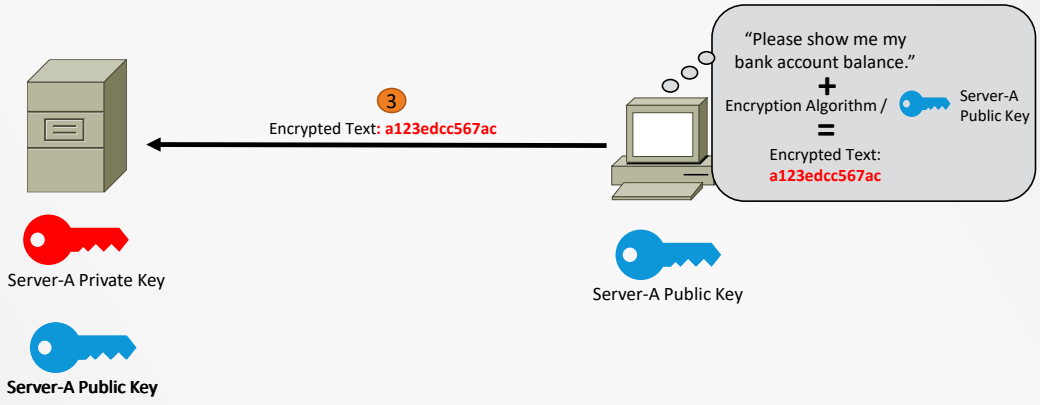


Asymmetric Keys



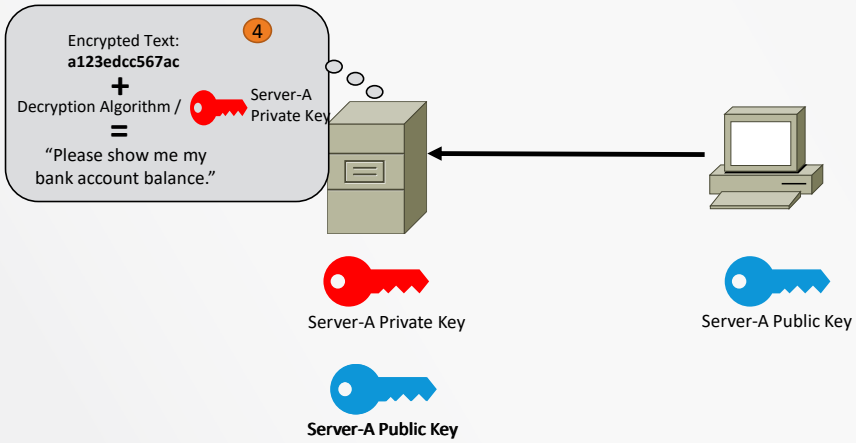
Step-2: Client utilizes Server's Public Key to encrypt a message destined for that Server.

Asymmetric Keys



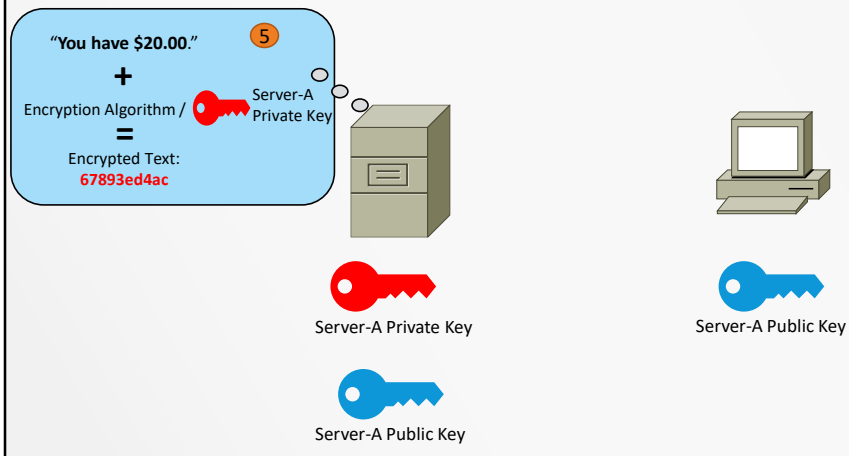
Step-3: Encrypted message transmitted to Server.

Asymmetric Keys



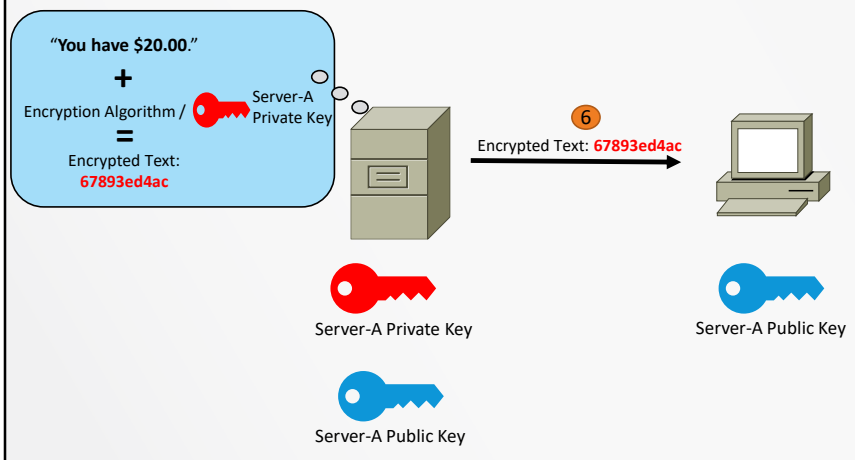
Step-4: Server utilizes its own Private Key to decrypt the message.

Asymmetric Keys

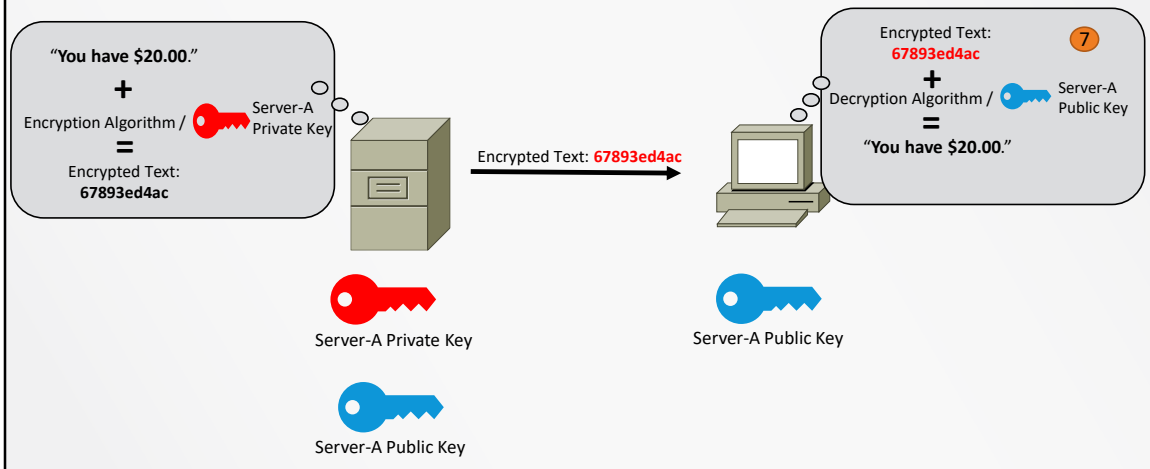


Step-5: Server generates a response and encrypts that response with its Private Key.

Asymmetric Keys



Asymmetric Keys



Step-7: Client decrypts Server's response with the Server's Public Key.

What Uses Asymmetric Encryption?

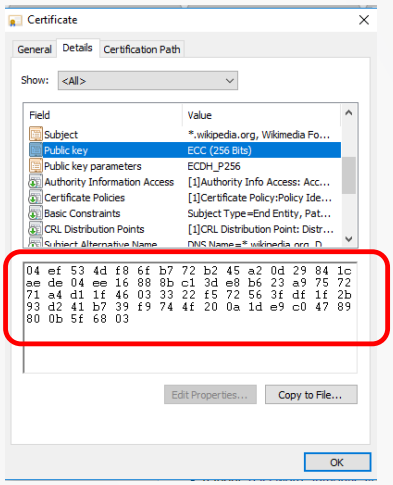
- Many protocols rely on Asymmetric Encryption;
  - SSH (RFC 4254)
  - OpenPGP (RFC 4880)
  - S/MIME (RFC 5751)
  - SSL/TLS (RFC 8446)

OpenPGP is an open and free version of the Pretty Good Privacy (PGP) standard that defines encryption formats to enable private messaging abilities for email and other message encryption.

-

Because asymmetric encryption is more computationally expensive than symmetric-key encryption, usually asymmetric-key encryption is utilized at the beginning of a conversation in order to create a secure, encrypted tunnel. Once that is established a shared-secret key can be either dynamically derived, or exchanged between the peers over that secure tunnel, so that successive encryption of data can utilize symmetric-key encryption.

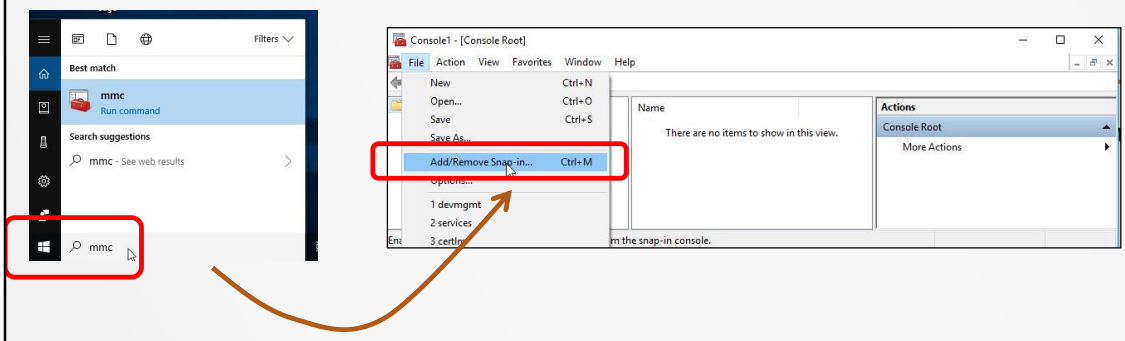
Viewing The Public Key



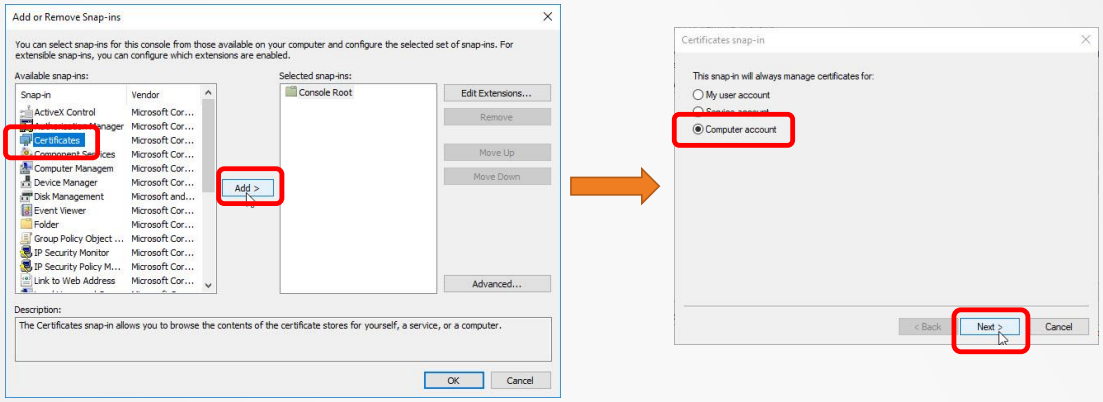
Private keys are very difficult to view as they are hidden deep in the OS of the machine that creates them.

Using MMC To View Public Keys

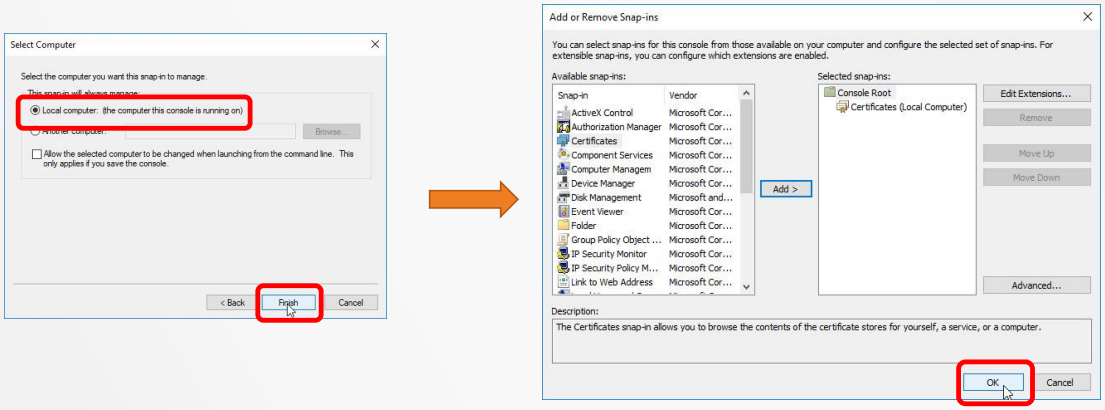
- Using the Microsoft Management Console (MMC)
- Type “mmc” from “Start/Run”



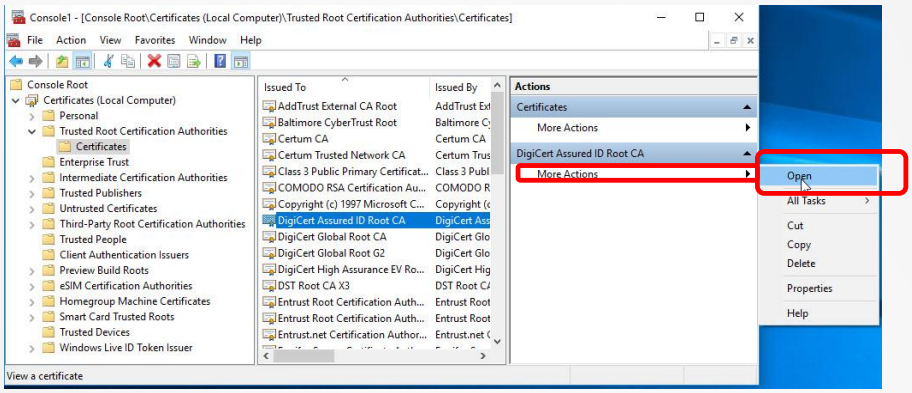
Using MMC To View Public Keys



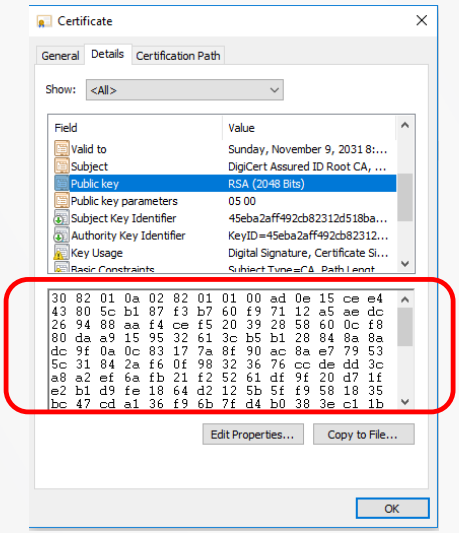
Using MMC To View Public Keys



Using MMC To View Public Keys



Using MMC To View Public Keys



Demonstration

- How to generate your own Public/Private Keypair