Crypto Series - Part 1 - An Overview

Brett - Team UnBlockable

March 25, 2020

Contents

What is Cyber Security?

And why should we care as App Developers?

- Cyber Security is the art and science of protecting systems, software, and data from unauthorized access, modification, or theft
- It has become a primary focus for every major technological industry. Many stories of its mishaps can be found on the news
- It should be a major component in the initial architecture and design of any system that we create
- CIA - Three things that we care about:
  - Confidentiality
  - Integrity
  - Authentication

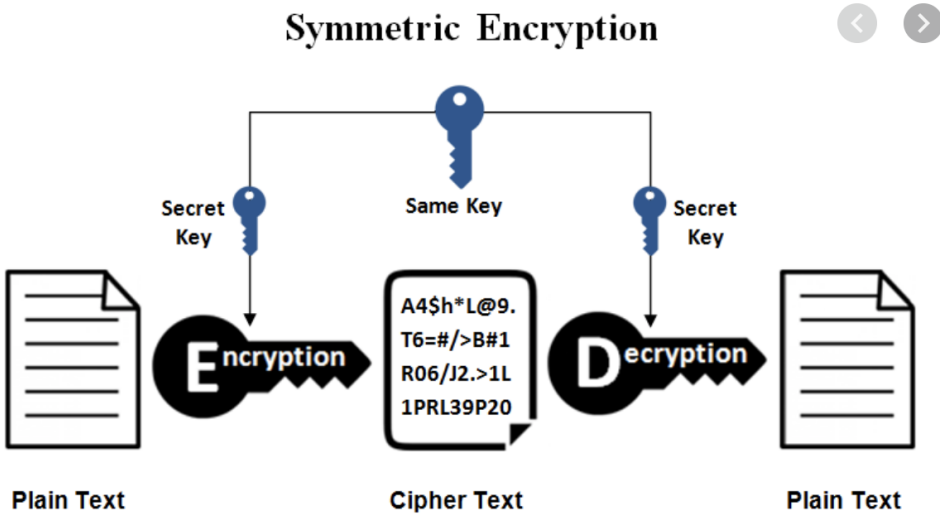
What is Cryptography?

- Cryptography is a compound word - built on crypto (secret) and graphy (writing)
- It is a science built on protecting information from people who should not see it
- There are many mechanisms and schemes for keeping data secret
- It is based on math and creativity

Symmetric Encryption

C, maybe I

- Symmetric Encryption allows Alice and Bob to exchange encrypted messages to each other, however, they must BOTH know the key in order to decrypt the message
- The encryption key and decryption key are the same for both Alice and Bob
- It is commonly used for securing a document with a password (i.e. password protected files)



Asymmetric Encryption

C, I, maybe A

- Also called Public Key Encryption
- Asymmetric Encryption introduces the idea of key pairs
- A key pair is comprised of both a public and a private key
- If Alice wants to talk to Bob using asymmetric encryption, Alice and Bob both will need their own key pair (4 keys total)

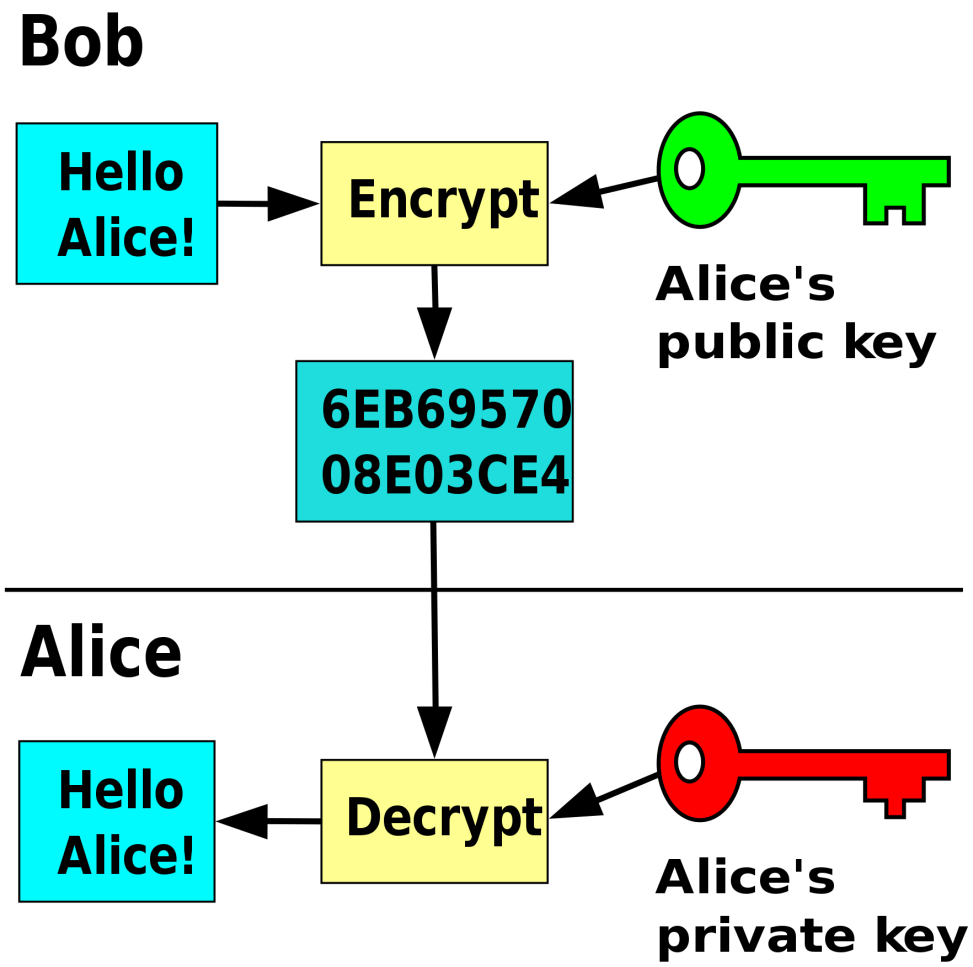
Asymmetric Encryption

Color Example

↓

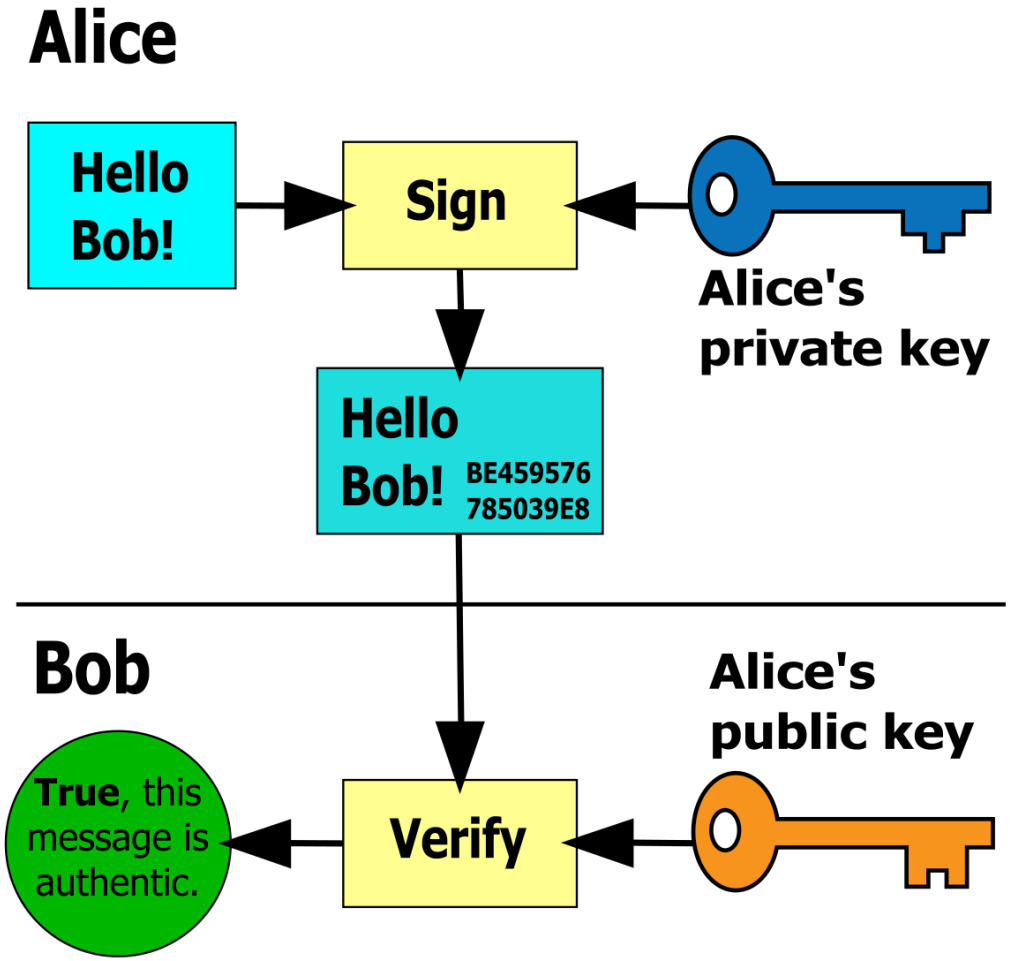
Asymmetric Encryption

Public Key Encryption - C, I



Asymmetric Encryption

Digital Signatures - A



What is wrong with this?

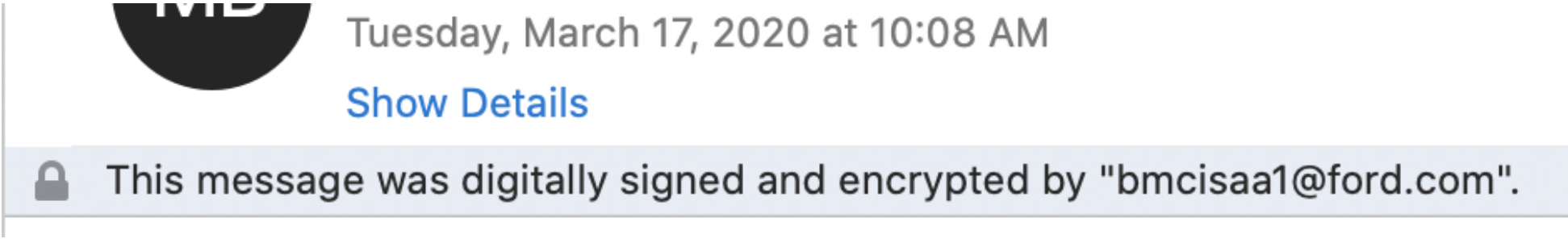
Digital Certificates

A

- Certificates provide validation. Think of it as a signature, but with authentication.
- In the previous example, a certificate could be sent with the public key to let Bob know that Alice's message really came from her by validating the following information. Information that is usually included:
  - Certificate owner's public key and its expiration date
  - Certificate owner's name
  - Certificate issuer's signature and name
- BASED ON TRUST IN THE THIRD PARTY

Email Signatures and Encryption

- The built-in Outlook app allows for digital signing and encrypting of emails
- Requires a Ford signed certificate and key pair (You will need to access a Windows machine to get this currently)



Email Signatures and Encryption

The Certificate and Key

View Certificate

- GlobalSign
  - Trusted Root CA SHA256 G2
    - Ford Motor Company - Enterprise Issuing CA01
      - BMCISAA1

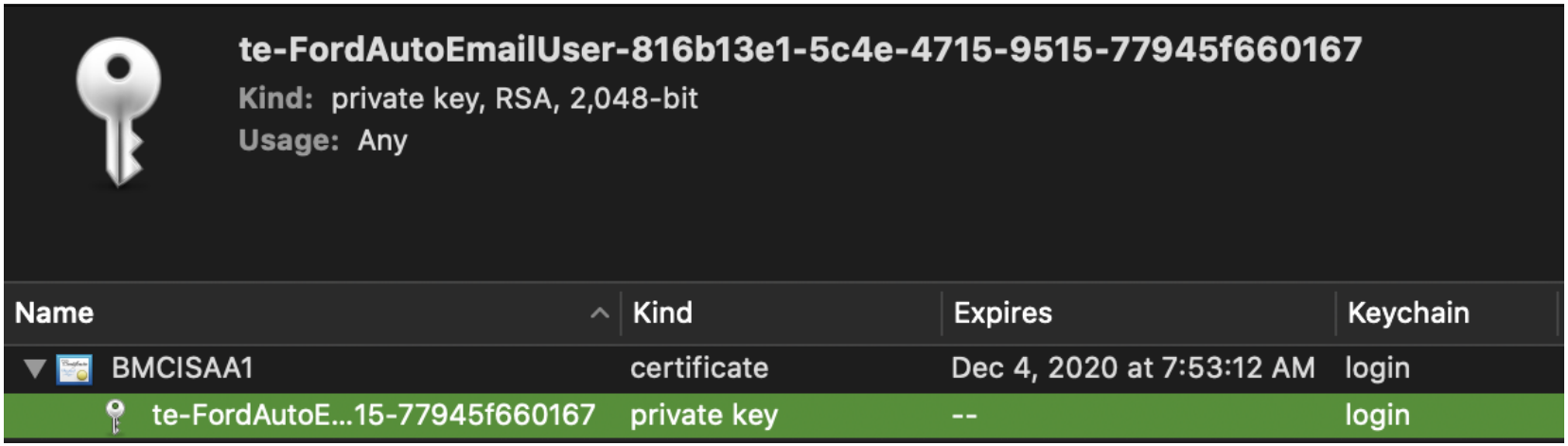
BMCISAA1
Issued by: Ford Motor Company - Enterprise Issuing CA01
Expires: Friday, December 4, 2020 at 7:53:12 AM Eastern Standard Time
This certificate is valid

Details

Subject Name	
Domain Component	com
Domain Component	ford
Domain Component	na2
Organizational Unit	USERIDS
Organizational Unit	GPO
Organizational Unit	Level0400
Common Name	BMCISAA1
Email Address	bmcisaa1@ford.com
Issuer Name	
Country or Region	US
Organization	Ford Motor Company
Locality	Dearborn
State/Province	Michigan
Common Name	Ford Motor Company - Enterprise Issuing CA01
Serial Number	66 94 0D EF 00 05 00 34 AB FB
Version	3
Signature Algorithm	SHA-256 with RSA Encryption (1.2.840.113549.1.1.11)
Parameters	None
Not Valid Before	Thursday, December 5, 2019 at 7:53:12 AM Eastern Standard Time
Not Valid After	Friday, December 4, 2020 at 7:53:12 AM Eastern Standard Time
Public Key Info	
Algorithm	RSA Encryption (1.2.840.113549.1.1.1)
Parameters	None
Public Key	256 bytes : D8 75 AA 9A 60 60 A9 61

Email Signatures and Encryption

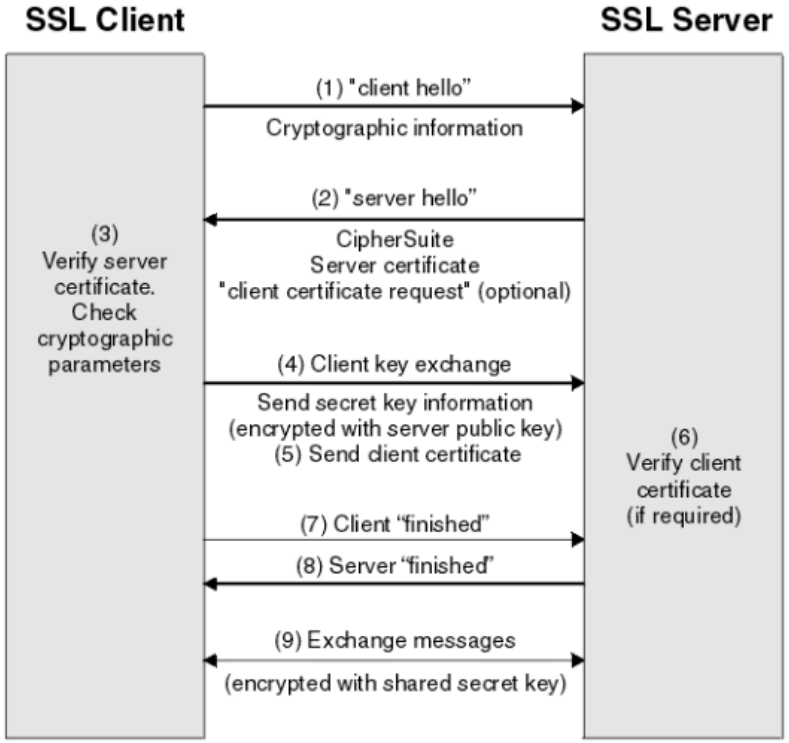
The Certificate and Key



TLS - Transport Layer Security

(Secure Socket Layering - SSL) - CIA

- Uses certificates that create a base of trust between a client and a server. Specifically, a certificate issued by a third party that verifies the identity of the server and its public key
- This ensures that all data remains private
- Consists of a Key Pair (Public/Private)
- Handshake:



TLS - Transport Layer Security

FordPass

- We needed to remove a Triple Data Encryption Standard (3DES) cipher
- It created a risk for a downgrade attack
- A recent FordPass example:

Merged | Restrict TLS connection to only 1.2 and 1.3 version #22
Changes from all commits | File filter... | Jump to...
0 / 7 files viewed | Review changes

42 network-utils-library/src/main/java/com/ford/networkutils/utils/RetrofitClientUtil.java | Viewed

```
            x509TrustManager = trustManager;
-           connectionSpecs = Arrays.asList(ConnectionSpec.MODERN_TLS, ConnectionSpec.CLEARTEXT); // enable TLS 1.0/1.1/1.2 for ht
+
+           connectionSpec = new ConnectionSpec.Builder(ConnectionSpec.RESTRICTED_TLS)
+                   .tlsVersions(TLS_1_2)
+                   .cipherSuites(
+                           CipherSuite.TLS_AES_128_GCM_SHA256,
+                           CipherSuite.TLS_AES_256_GCM_SHA384,
+                           CipherSuite.TLS_CHACHA20_POLY1305_SHA256,
+                           CipherSuite.TLS_AES_128_CCM_SHA256,
+                           CipherSuite.TLS_AES_256_CCM_8_SHA256,
+
+                           CipherSuite.TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256,
+                           CipherSuite.TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256,
+                           CipherSuite.TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384,
+                           CipherSuite.TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384,
+                           CipherSuite.TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256,
+                           CipherSuite.TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256,
+
+                           // Note that the following cipher suites are all on HTTP/2's bad cipher suites list. We'll
+                           // continue to include them until better suites are commonly available. For example, none
+                           // of the better cipher suites listed above shipped with Android 4.4 or Java 7.
+                           CipherSuite.TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA,
+                           CipherSuite.TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA,
+                           CipherSuite.TLS_RSA_WITH_AES_128_GCM_SHA256,
+                           CipherSuite.TLS_RSA_WITH_AES_256_GCM_SHA384,
+                           CipherSuite.TLS_RSA_WITH_AES_128_CBC_SHA,
+                           CipherSuite.TLS_RSA_WITH_AES_256_CBC_SHA)
+                   .build();
        }
```

Proxies

How does Charles Proxy Work?

- We need to perform a Man In The Middle Attack
- Charles intercepts the network call from the phone and then will act as the client. This allows the call to be started with Charles' certificate and key which allows it to decrypt the response.
- Because the phone will now be receiving the response from Charles, and not some other trusted source, we need to install its certificate in the phone.
- We then need to go to chls.pro/ssl to download that certificate, install it, and force the phone to trust it. The phone then trusts that it's connected to a safe server and will perform normally.

Certificate Pinning

How do we prevent MITM attacks? What could help?

- Certificate Pinning keeps a copy of one of the levels of certificates that you *should* expect from your server and compares what it gets versus what it should expect
- Newer versions of this use hashes instead of using entire certificates
- It can be done with the *CertificatePinner* class in OkHttp or *AFSecurityPolicy* in AFNetworking
- While this is good in theory, it can be dangerous to use in production. Care must be taken to have communication between the server and mobile teams. Failure to update certificates would essentially brick the app for your users, until a new version could be downloaded

The End