CRYPTOGRAPHY

An Introduction to Cryptography and the Public Key Infrastructure

by Michael Gates, Sales Engineer, REDCOM Laboratories, Inc.

1. Introduction

Like it or not, we are living in a virtual world and using the internet for everyday tasks has become commonplace. We can visit a virtual branch of our bank and pay our bills, stream the latest episode of our favorite television program, or even take our credit card on a spending spree at a virtual mall. The internet is a major component of our lives today and provides an extremely convenient service. We sometimes need to be cautious when using that service, however. What if your bank account or credit card information was available to others during your stroll through the virtual world?

While this document won't provide specific details on protecting your bank account, the methods used to protect this type of information are applicable to protecting other types of data. Much like the security used by a web browser to protect your bank account information the same methods can be used to protect other types of information, such as the signaling and media transmitted during a voice over internet protocol (VoIP) call.

Without a doubt, the methods and tools used to secure information that is transmitted across a network are complex subjects. Entire textbooks are devoted to each of these areas. This document is not intended to provide a comprehensive study of these subjects. Rather, a brief overview with enough information to understand the fundamentals of the subject is the goal here. Sources of additional reading will be provided at the end of the document should you wish to dig deeper into these topics.

Contents

2. Cryptography

Before diving head first into the subject of a public key infrastructure (PKI), it is necessary to first understand the basics of cryptography. The Merriam-Webster dictionary defines cryptography as follows:

1: secret writing
2: the enciphering and deciphering of messages in secret code or cipher; also: the computerized encoding and decoding of information

Simply stated, cryptography is the means for taking a readable message (aka plain text) and converting it into an encoded message (aka cipher text) or vice versa. The algorithm used to encode/decode the message is known as a cipher. The act of encoding is referred to as encryption, while the act of decoding is decryption. The intent of this process is to render the plain text into an unreadable format to all but a select group of people who possess the appropriate key to unlock the secret.

To illustrate the basics of cryptography, let's take a look at an example. One of the earliest known uses of cryptography was a cipher used by Julius Caesar that eventually became known as the Caesar cipher. Today this type of encryption is commonly known as a substitution or shift cipher. Using the English alphabet, we can easily demonstrate how this simple cipher works. This cipher works by substituting each plain text letter in a message with a cipher text letter. Imagine the letters of the alphabet laid out in two rows. We can designate the top row as plain text and the bottom row as the cipher text. This is illustrated in the diagram shown below.

Plain Text →	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Cipher Text →	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z

The Caesar cipher works by simply shifting the letters by a specified number of positions (shift cipher). For our example, let's use a shift of five. All letters in the cipher text row are shifted by five positions. As a result, we end up with the cipher shown below.

Plain Text →	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Cipher Text →	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U

To use this cipher, we now replace each plain text letter with its corresponding cipher text letter (substitution cipher). For example a plain text letter A is replaced with a cipher text letter V, a plain text letter B is replaced with a cipher text letter W, and so on. Using this cipher, let's encrypt the plain text word CIPHER.

Plain Text	C	I	P	H	E	R
	↓	↓	↓	↓	↓	↓
Cipher Text	X	D	K	C	Z	M

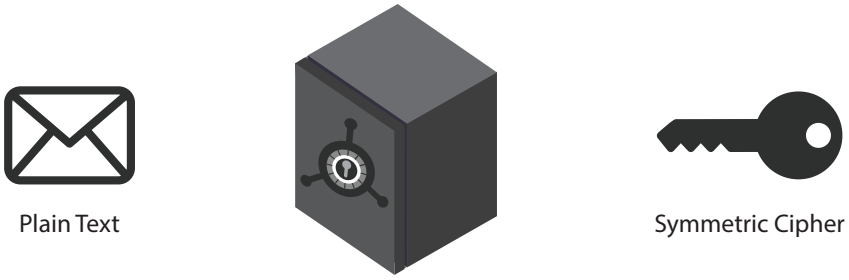
Using a Caesar cipher of five, we can see that the plain text word CIPHER is encoded to become the cipher text word of XDKCZM. To most people, this word is unrecognizable. But, since we possess the key to unlock the hidden word, we're able to decipher its meaning. In order to retrieve the plain text word, simply take each cipher text letter and map back up to its corresponding plain text letter.

Following the above example provides a basic understanding of how plain text can be encoded into cipher text. Today's cipher algorithms are obviously much more complex than using a simple shift cipher. The mathematics of how these complex ciphers operate is well outside of the scope of this document, however.

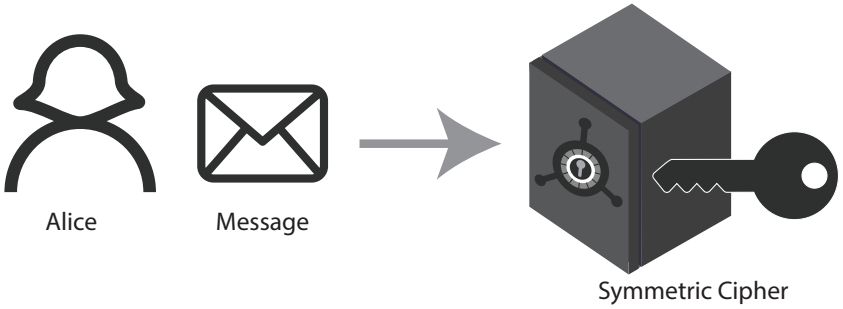
There are two primary methods of cryptography, symmetric and asymmetric. These terms refer to how the encryption and decryption ciphers relate to each other. Next, we will explore both methods in the upcoming sections.

2.1 Symmetric Cryptography

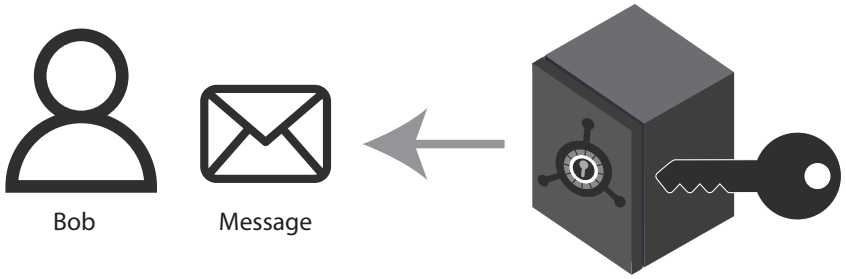
In symmetric cryptography the same cipher is used to both encrypt the plain text as well as decrypt the cipher text. The Caesar cipher discussed in the previous section is an example of symmetric cryptography. The symmetric cryptography process can best be described with an analogy. First, imagine that a letter represents plain text, a lockable safe represents cipher text, and a key to that safe represents a symmetric cipher as shown in the illustration below.



Now, further imagine that Alice and Bob wish to exchange messages confidentially. Once Alice has created her plain text message (letter), she can encrypt it by placing it into the safe and using her key (symmetric cipher) to lock it (cipher text).



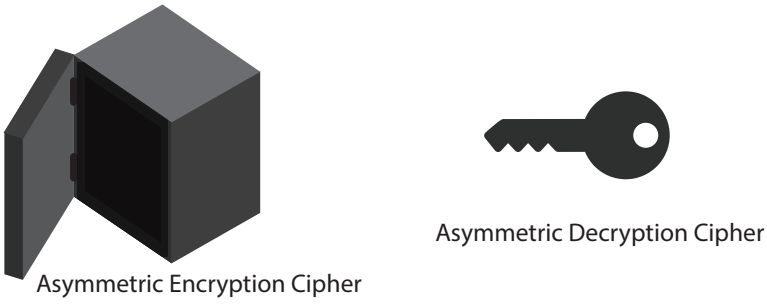
Now that she has done so, she can send the encrypted text to Bob and be confident that it remains confidential. Once Bob receives the cipher text (safe) he can decrypt the message by using the symmetric cipher (key) to decrypt the encrypted text and retrieve Alice's plain text message (letter)



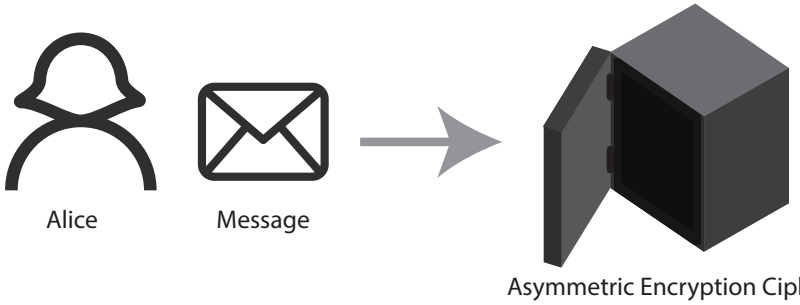
Since both Alice's and Bob's key are identical, they used symmetric cryptography to communicate confidentially. Symmetric ciphers that are commonly in use today are the triple Data Encryption Standard (3DES) and the Advanced Encryption Standard (AES).

2.2 Asymmetric Cryptography

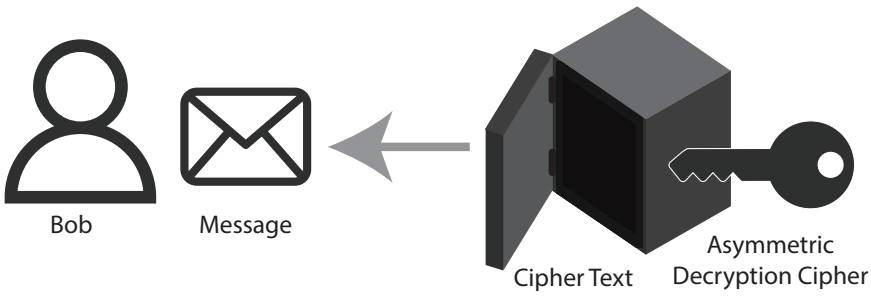
Asymmetric cryptography differs from symmetric cryptography because the cipher used for encryption and the cipher used for decryption are different. Asymmetric ciphers are created and work together in pairs. Anything encrypted by an asymmetric cipher can only be decrypted by its pair. The asymmetric cryptography concept can be illustrated by using a modified version of the analogy used for symmetric cryptography. For asymmetric cryptography, the safe represents the half of a cipher pair that is used for encryption while the key represents the other half of a cipher pair that is used for decryption. Only the key (decryption cipher) that is associated with the lock on the safe (encryption cipher) can be used to open it. No other key can be used to open the safe.



Our friends Alice and Bob again wish to communicate confidentially. This time, however, they will be using an asymmetric cipher pair. Alice begins the process to encrypt her message by placing it into the safe and closing it up. She then sends the safe to Bob.



Once Bob receives the safe, he is able to decrypt and retrieve Alice's message since he has the key to unlock it.



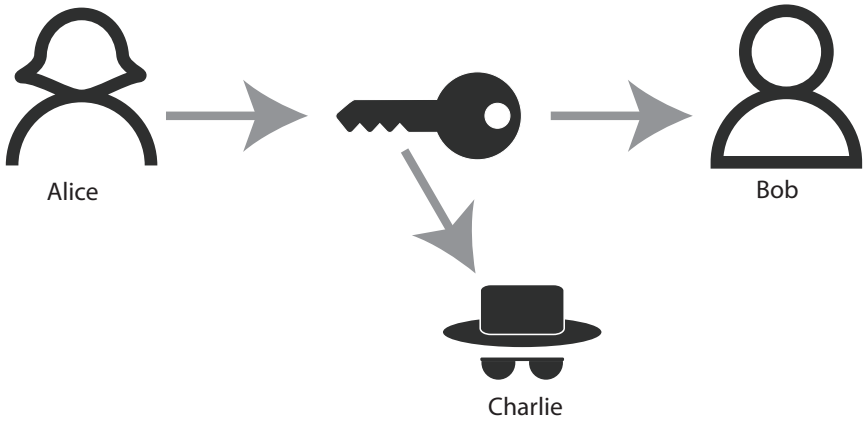
Here, the cipher that Alice used to encrypt the message (safe) was not the same cipher that Bob used to decrypt the message (key). The two most common asymmetric ciphers in use today are the Rivest, Shamir, and Adleman (RSA) cipher – so named for its creators – and the Elliptic Curve Diffie-Hellman (ECDH) cipher.

2.3 Vulnerabilities

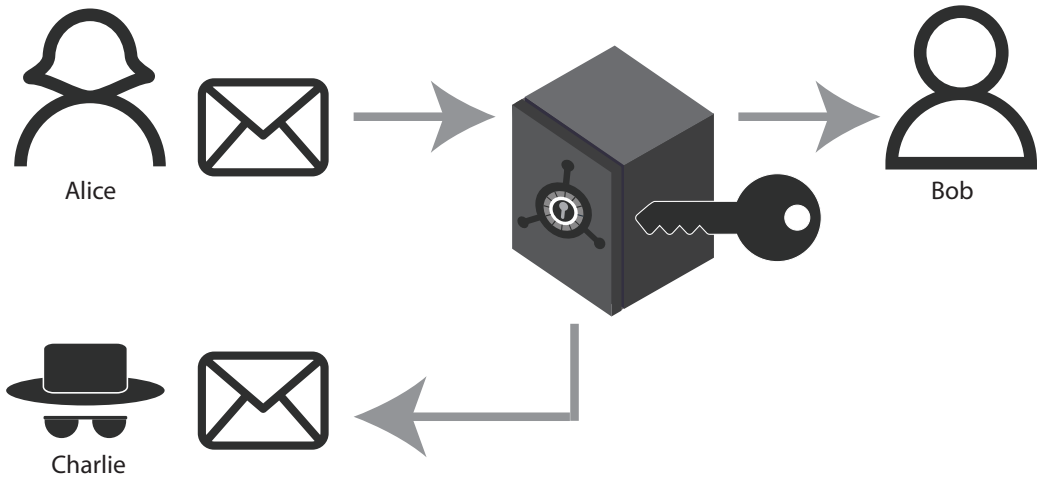
In reality, a perfectly secure encryption/decryption process does not exist. Let's consider some scenarios that present opportunities for nefarious behavior.

2.3.1 Symmetric Vulnerabilities

Symmetric ciphers can be somewhat difficult to distribute. As we saw in the example from section 2.1, after Alice sends her encrypted message to Bob he needs a copy of the key to unlock it. So, how does he get it? Alice can send it to Bob, but it becomes vulnerable to interception. Imagine that Charlie is monitoring all communications between Alice and Bob with malicious intent and, therefore, has an opportunity to intercept the key when it's sent.

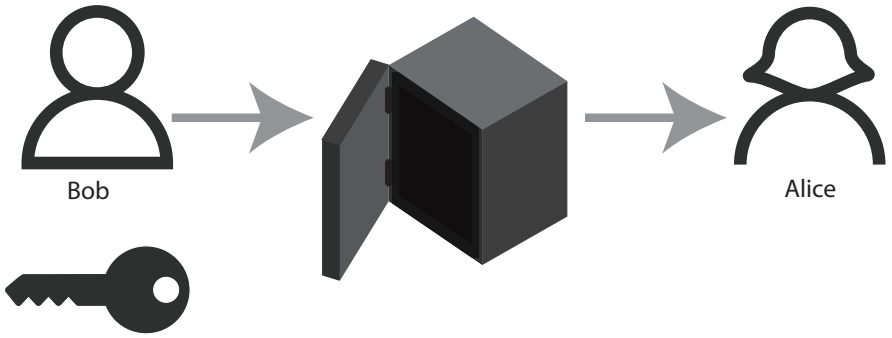


Charlie now has the cipher he needs to decrypt all communications between Alice and Bob. As he intercepts each encrypted message, he can use the illicit key and retrieve the message.



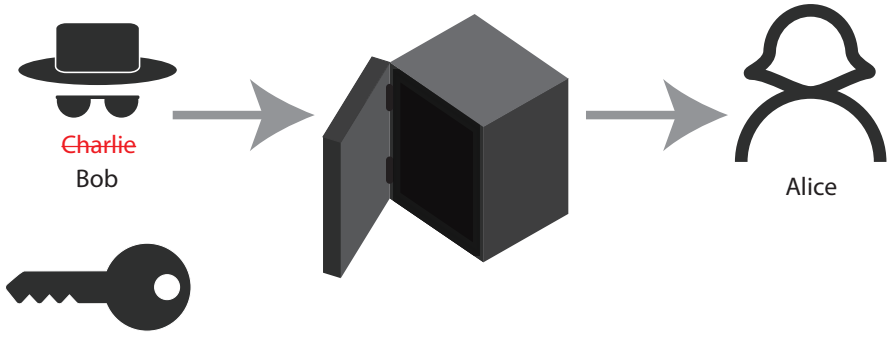
2.3.2 Asymmetric Vulnerabilities

In reality, a perfectly secure encryption/decryption process does not exist. Let's consider some scenarios that present opportunities for nefarious behavior.

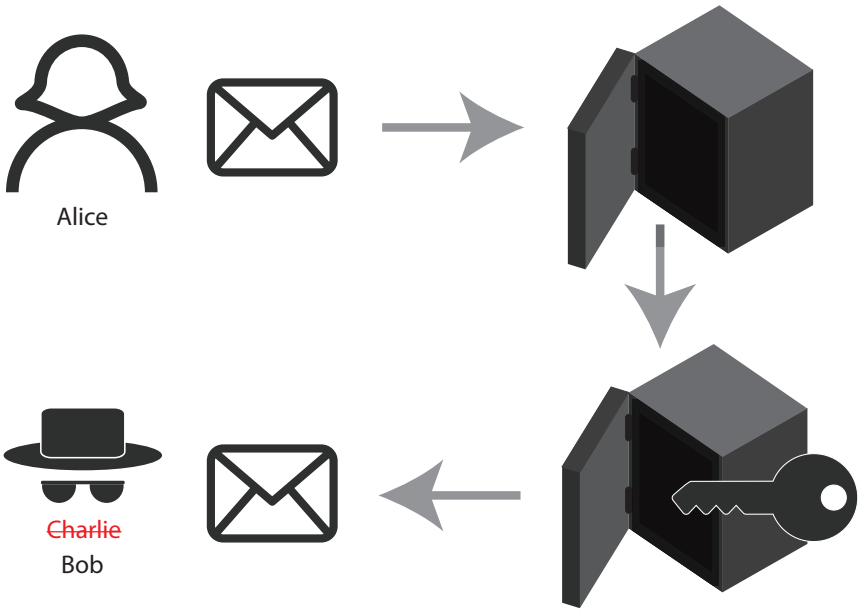


Alice can use that cipher to encrypt messages and send them to Bob as described in section 2.2. Should Charlie intercept the cipher or the encrypted messages, he will be unable to decrypt them since he does not have the paired cipher that Bob held onto.

Charlie, however, can take a different approach. What happens if Charlie impersonates Bob? He can generate his own key pair and send one of them to Alice claiming to be Bob.



Alice, believing that she is communicating with Bob, will now use that cipher to encrypt messages intended for Bob and send them to Charlie. Charlie has the paired cipher and can easily decrypt and retrieve them.



3 Public Key Infrastructure

With the current state of affairs in the world today, the terms 'cyber security' is becoming more widely known. Cyber security is the process for protecting information from unauthorized access, use, disclosure, disruption, modification, or destruction. In order to achieve the goals of cyber security, a multi-faceted approach is used and a public key infrastructure (PKI) plays an important role.

A PKI is comprised of a blend of technology and policy in order to facilitate confidential communications between two or more entities. In order to accomplish this, several different components are necessary. These components are broken down and discussed over the next few sections.

3.1 Digital Certificates

A digital certificate is data used by an asymmetric cipher during the encryption/decryption process. Within a PKI, policy assigns very specific functions to the certificates. The intent of this policy is to aid in the distribution problem described in section 2.3.1. When an asymmetric certificate pair is created, one is designated as private and the other as public. The private certificate (sometimes referred to as a private key) is not to be distributed and kept only by the entity that created it. The public certificate can be freely distributed to any entity between which confidential communication is desired. This process is generally referred to as public key cryptography.

For example, Alice can create her digital certificate pair. She will keep the private key to herself. She can then freely distribute the public certificate to others that she wishes to confidentially communicate with, such as Bob. When Bob wants to send a confidential message to Alice, he will first encrypt it with her public key. The only thing that can then decrypt that message is Alice's private key. Since she has kept it to herself, she can feel confident that the message was kept confidential.

3.2 Digital Signature

In the example above, Bob encrypts a message with Alice's public key and then Alice decrypts it with her private key. While this is the most common use of public key cryptography, the reverse can also be true. In other words, Alice can encrypt a message with her private key and Bob can decrypt it with her public key. Since Alice is the only person that has her private key, this can be considered as proof that Alice originated the message. She has, in effect, digitally signed the message.

This method should not be used to send confidential messages to Bob, however. Since Alice's public certificate was freely distributed, anyone that has a copy of it will be able to decrypt and retrieve the original message. If Alice needs to send a confidential message to Bob, she should get a copy of Bob's public certificate.

3.3 Trust

Recall from section 2.3.2 that Charlie could simply impersonate Bob. The simple fact is that an individual who possesses what they claim to be Bob's private key is not sufficient proof that they are actually Bob. So, how do we prove beyond any reasonable doubt that Bob is indeed Bob? Within a PKI, this is accomplished with a system of trust.

To provide trust, a certificate authority (CA) is used. The CA is responsible for issuing and maintaining public certificates. Prior to issuing a public certificate, the CA will verify the identity of the individual requesting the certificate. How this is accomplished depends on the type of entity making the request. For example, if a person is requesting a certificate the CA may check employment records, social security records, or even a birth certificate. Once the CA has issued a public certificate, other users can decide to trust the CA. They are then trusting that the CA has in fact validated the user's identity and, by extension, are trusting that the individual is who they claim to be.

The process can best be explained with a practical example. When Bob wants to join a trusted CA, he will forward a request. The CA will take steps to verify Bob's identity to ensure that he is indeed Bob. Once the CA is satisfied with the verification, it will digitally sign Bob's public certificate. Now, when Bob wishes to communicate with Alice, she can verify his identity by using the CA's public certificate (generally referred to as the root certificate) to validate that the CA has digitally signed Bob's public certificate. If this process is successful, and Alice trusts the CA that Bob uses, then Alice can be reasonably sure that the public certificate she is using does belong to Bob.

3.4 Certificate Validation

Digital certificates do not have an indefinite life span. There are two ways that certificates can become invalid, they can expire or become compromised. When a certificate is issued, the CA will require that it be renewed periodically so the owner's identity can be re-verified. As a result, certificates are issued with an expiration date (typically three years from date of issue). It is also possible that an unauthorized user has compromised the certificate by gaining access to the private key. There are many ways that this can occur. Perhaps Charlie breached physical security, or used social engineering and phishing attacks, to gain access to Bob's computer and steal his private key. In this case, Bob must notify the CA that his certificate has been compromised so it can be revoked.

It is, therefore, incumbent on the CA to maintain a list of the current status for each certificate. Then, whenever a public certificate is received during an attempt to establish confidential communications, its current status should be validated against this list. There are currently two methods employed for performing certificate validation, using a certificate revocation list (CRL) or the Online Certificate Status Protocol (OCSP).

A CRL is simply a list that contains all certificates a CA has issued and their current status. This list is updated and published periodically by the CA. Entities wishing to validate certificates issued by the CA will then need to download the CRL. At that point, validating a certificate is a simple lookup in the list to determine its status.

OCSP negates the need for every entity downloading its own copy of the CRL. With OCSP, a central server will download the CRL and maintain it locally. A device that wishes to validate a certificate simply sends an OCSP request to the server (known as an OCSP responder) with the certificate information. The OCSP responder then verifies the current status of the certificate and replies to the request with the current status information for the certificate.

While either method of certificate validation is acceptable, using OCSP is preferred. Since a CRL contains the status of every certificate issued by a CA, they can be quite large in size. Also consider that there are many CAs in use around the world. Each CRL needs to be downloaded periodically, and each individual device performing certificate validation will need to download its own copy of the CRL. A considerable amount of bandwidth will be needed simply to periodically download CRLs. If an OCSP responder is implemented, only a single copy of each CRL needs to be downloaded to a server that is providing validation services for many endpoints.

3.5 Goals of a Public Key Infrastructure

There are three primary goals of the services that a PKI provides: confidentiality, authentication, and non-repudiation. Each of these goals are discussed in the upcoming sections.

3.5.1 Confidentiality

Confidentiality allows authorized users access to data while blocking unauthorized disclosure to others. Confidentiality is achieved with the encryption methods discussed previously. One often misunderstood concept is which type of cryptography a PKI actually uses. In reality, due to the vulnerabilities discussed in section 2.3, a PKI utilizes both symmetric and asymmetric cryptography.

Symmetric keys are typically smaller and, as a result, are more efficient and encrypt/decrypt data faster. For this reason, they are preferred for most encryption scenarios. Key distribution is a concern, however. This is where asymmetric cryptography comes into play. Asymmetric cryptography can be used to freely distribute the public keys and establish a trusted encrypted path over which symmetric keys can be confidentially distributed.

3.5.2 Authentication

Authentication is the practice of verifying the identity of the other party involved in a confidential communication. Authentication is accomplished with the use of digital signatures and the policy of trust. The process of

authentication is a very common practice with confidential digital communications. For example, consider using a web browser to connect to your bank's website to view account information. Have you ever noticed the lock symbol that appears when you log into your account? This is an indication that the website has been authenticated and encrypted communications have been established.

When you connect to your bank's website, the bank's public key is provided along with a digitally signed message. The web browser will decrypt the digitally signed message with the bank's public certificate and, if the CA is trusted, authenticate the identity of the bank. The browser then uses the bank's public key to encrypt and exchange a symmetric key used to communicate confidentially with the bank.

The above paragraph illustrates an example of one-way authentication. Here, your web browser authenticated the identity of the bank's website. It is also possible that both parties involved authenticate each other. This is generally referred to as mutual authentication.

3.5.3 Non-repudiation

In order to define non-repudiation, it may be better to first define repudiate. According to Merriam-Webster, the definition of repudiate includes:

1: to reject as untrue
2: to refuse to acknowledge

So, non-repudiation is the ability to prevent the denial that something happened. In simpler terms, it is proof that a transaction occurred thereby preventing an attempt to deny that it occurred in the first place. Digital signatures can also provide this service. If a party provides a digital signature as proof of their identity, the other party can also use that as proof that the communication occurred.

3.6 The Process – Putting it All Together

Now that we have been introduced to both the technology and the policies employed by a PKI, let's take a holistic look at the PKI process itself. In this section, we will walk through the entire process from joining a PKI through establishing an encrypted connection.

3.6.1 Obtaining a Certificate Pair

In order to join a PKI a key pair must first be generated for the individual or device. There are several steps in this process; a certificate pair must be generated, a request for authorization must be sent to the CA, the CA must verify the identity of the individual or device, and finally the CA will authorize the individual or device and issue a public certificate. This entire process is initiated with the generation of a certificate signing request (CSR).

When a CSR is generated two things are created: the CSR itself and the private key. As mentioned previously, tight control must be maintained over the private key. The CSR will then be forwarded to the CA for authorization. Once the CA has verified the identity of the entity making the request, the CA will digitally sign the CSR with

its private key. This file is then returned to the requestor and becomes their public certificate.

A CSR along with the public and private keys use the X.509 standard. This standard defines the structure and contents of a certificate. The X.509 certificates and keys are generally kept in a compressed format. An example of X.509 format as well as compressed versions of a public certificate are provided in sections A1.1 and A1.2 respectively.

3.6.2 A Protocol for Confidential Communications

Devices on a network use many different protocols to communicate. The choice of protocol depends on the topic at hand. A common choice for establishing a path for confidential communications is the transport layer security (TLS) protocol. TLS is a successor, and therefore very similar, to the secure sockets layer (SSL) protocol. Following a TLS dialog as it negotiates a connection will help to put all of the pieces together and illustrate the topics discussed in this document.

The TLS protocol is used to authenticate the endpoints and establish an encrypted path. The endpoint that initiates the negotiation (party A) will always adopt the role of client, while the other endpoint (party B) will become the server. Should another negotiation take place between the same two parties at a different time with party B initiating the process, then party B will be the client while party A the server. The roles are related only to the initiator, not the devices themselves.

The steps below follow a TLS dialog between two parties attempting to authenticate each other (mutual authentication) and establish an encrypted path to carry out a confidential conversation. The OCSP protocol is employed to perform certificate validation to ensure that the certificates haven't expired or been compromised and revoked as a result. These steps correspond to a ladder diagram to highlight each of the steps in the process. The ladder diagram is shown in section A1.3.

1. A TLS negotiation is always initiated with a CLIENT HELLO message. This message contains a list of cipher supported by the client along with a random number (random1) that will be used during the generation of the symmetric key.
2. The second party responds with a SERVER HELLO message. This message includes the encryption cipher chosen (from the client's list) and a second random number (random2) to be used during the generation of the symmetric key.
3. The server sends a CERTIFICATE message that contains its public certificate and the public certificate of the certificate authority.
4. The client then authenticates server's public key by checking the digital signature with the CA's public certificate. The client then validates the server's public key with an OCSP responder.
5. Since mutual authentication is being performed, the server sends a CERTIFICATE REQUEST message to request the client's public certificate.
6. The server then indicates it is done with its requests by sending a SERVER HELLO DONE message.
7. The client sends a CERTIFICATE message that contains its public certificate and the public certificate of its

certificate authority.

8. The server then authenticates and validates the client's public key.

9. The client generates a third random number (secret), referred to as the pre-master secret, encrypts this with the server's public key, and sends it to the server with a CLIENT KEY EXCHANGE message. The server can then decrypt it with its private key.

10. The client and server each use both random numbers and the pre-master secret to generate the symmetric encryption key.

11. The client and server then exchange CHANGE CIPHER messages with test data as an indication that the key generation was successful and the symmetric key should be used for all further communication.

12. The client and server are now communicating over an encrypted path. Since asymmetric cryptography was used to conceal the symmetric key, it was confidentially distributed between the two parties.

Appendices

A1 Supporting Information

A1.1 X.509 Format

```
Certificate: ca2.cer
  Data:
    Version: 3 (0x2)
    Serial Number: 5 (0x5)
    Signature Algorithm: sha1WithRSAEncryption
    Issuer: C=US, O=U.S. Government, OU=DoD, OU=PKI, CN=DoD JITC Root CA 2
    Validity
      Not Before: Jul 15 03:31:31 2005 GMT
      Not After : Jul  4 03:31:31 2030 GMT
    Subject: C=US, O=U.S. Government, OU=DoD, OU=PKI, CN=DoD JITC Root CA 2
    Subject Public Key Info:
      Public Key Algorithm: rsaEncryption
        Public-Key: (2048 bit)
        Modulus:
          b4:48:ca:63:29:78:43:a1:94:65:ac:9c:0a:d2:77:2b:
          83:30:df:2f:b3:92:d4:cd:ee:97:8d:ea:47:56:05:ea:
          04:f4:31:d8:05:9b:f7:1d:81:38:87:e3:4a:a0:88:ca:
          5f:06:38:1c:a7:24:bc:1a:c8:66:29:51:12:46:49:c9:
          df:2d:93:12:5e:bb:7a:48:73:e8:7f:68:27:97:e8:6a:
          03:bd:52:b1:bc:3e:8e:d3:3c:b4:00:e5:7e:06:93:db:
          59:28:d6:26:26:f3:c9:da:9a:56:f7:15:16:d3:cf:81:
          b6:70:1b:07:21:a0:f4:0e:19:03:d8:35:6d:3d:42:61:
          5b:86:9b:56:24:d0:60:f7:f8:fe:7e:3d:eb:80:5d:0b:
```

6e:8e:eb:a5:ad:2a:c7:ed:5a:c8:15:79:07:7d:0f:57:
9f:0a:92:9f:32:cf:e6:a6:dc:20:4c:42:6a:22:f2:ae:
be:15:7e:fe:b0:33:19:f4:e4:0d:49:b8:d7:8f:14:d7:
f1:ee:11:82:6f:23:ef:a1:d2:96:69:4b:ec:33:40:49:
01:e5:02:66:f2:ba:e9:ed:b7:87:06:22:c3:e9:86:74:
3a:70:13:82:79:52:a0:03:f8:e8:e2:ee:2d:f1:e5:5f:
c7:67:dc:dd:0b:fb:17:c8:31:cd:f0:3a:86:ad:4c:3b

X509v3 extensions:

X509v3 Subject Key Identifier:

F9:E0:3F:87:56:FF:D2:21:80:BA:3D:13:7E:C5:4F:54:B0:DF:BC:02

X509v3 Key Usage:

Digital Signature, Certificate Sign, CRL Sign

X509v3 Basic Constraints: critical

CA:TRUE

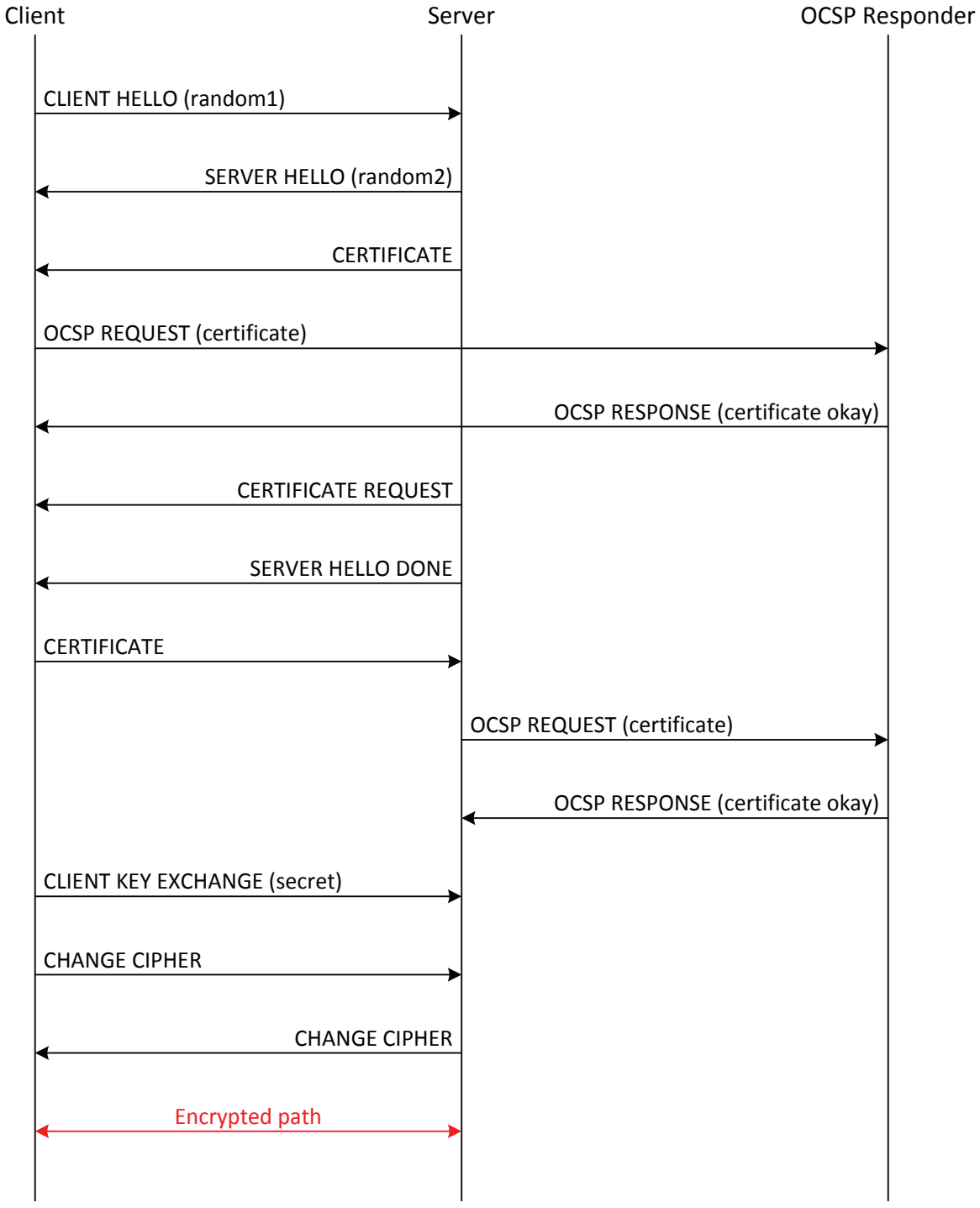
Signature Algorithm: sha1WithRSAEncryption

3f:b0:bc:e5:ff:fb:9e:05:8c:e3:dc:2e:b0:45:be:f8:
b1:4c:30:dd:8c:9a:f6:4e:4d:cb:77:f5:ee:2e:58:b1:
b3:b3:00:1b:ff:51:c6:a7:74:16:c5:99:f6:27:ac:05:
f0:ed:9c:c2:a6:68:17:81:ca:59:46:7b:71:20:75:5e:
25:4f:d6:cc:58:44:06:e3:2a:5c:07:5c:d7:d8:3e:6e:
4c:e1:fc:e1:39:7c:43:83:04:ca:55:cc:b5:2c:bd:14:
0b:14:1a:75:d8:63:fa:d0:05:0b:5e:48:de:f2:4d:61:
d0:d5:f1:e0:43:6d:28:56:ce:60:ba:db:d3:3c:09:54:
ca:93:44:68:27:9d:83:3d:77:37:ea:c4:0d:37:75:79:
ef:09:56:5a:67:95:65:63:03:2f:ab:27:f1:4e:df:29:
48:8f:a2:e2:f3:33:7a:b3:68:8d:9a:ae:83:43:d7:d6:
d0:3b:86:32:a4:ee:03:d2:ed:b0:fa:d9:fc:24:a1:56:
7a:c1:3f:97:c6:d6:74:20:57:2a:4b:13:d5:9b:42:a0:
db:44:db:b5:f1:a8:1e:c1:0a:71:fd:9a:a3:53:f1:12:
f2:b1:98:ef:5c:5b:ae:65:21:6f:7a:9a:ed:a9:2a:32:
4d:6f:e1:66:5b:a4:40:8d:a0:c9:5c:51:e4:37:cb:8f

A1.2 Example Compressed Certificate

-----BEGIN CERTIFICATE-----
MIIFpjCCBQ+gAwIBAgIBYzANBgkqhkiG9w0BAQUFADCB0jELMAkGA1UEBhMCVVMx
ETAPBgNVBAgTCE5IdyBZb3JrMQ8wDQYDVQQHEwZWaWN0b3IxIjAgBgNVBAoTGVJF
RENPTSBMYWJvcmF0b3JpZXMsIEluYy4xHjAcBgNVBAsTFUNlcnRpZmljYXRlIEF1
dGhvcml0eTE4MDYGA1UEAxMvUkVEQ09NIExhYm9yYXRvcmllcywgSW5jLiBDZXJ0
aWZpY2F0ZSBBdXRob3JpdHkxITAfBgkqhkiG9w0BCQEWEnN1cHBvcnRAcmVkY29t
LmNvbTAeFw0xMjAxMTExNjQxMzZaFw0xNzAxMTAxNjQxMzZaMIHWMQswCQYDVQQG
EwJVUzERMA8GA1UECBMITmV3IFlvcmsxDzANBgNVBAcTBlZpY3RvcjEiMCAGA1UE
ChMZUkVEQ09NIExhYm9yYXRvcmllcywgSW5jLjEwMC4GCgmSJomT8ixkAQETIDBj
NTZkZjJkYmM5MjRmNTYyNTJjMWJjYjE5NmIxMDFiMRcwFQYDVQQLEw5MYWItU0xJ
Q0UgMjEwMDESMBAGA1UEAxMJMTAuMTAuMi4xMSAwHgYJKoZIhvcNAQkBFhFtZ2F0
ZXNAcmVkY29tLmNvbTCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBAOby
/RWu/PCCop0Vd++vlDnt66BzBrktvOJUkzmFBDnD+Cub6JP0SHoRNbfPKbXXtJwV
mFR506qK4inI41Bwxc3fal8KV8euKs9YCiidflqrFJX5oVFQdKinVWwFCfHxRUdy
Z5OB2O38YVSQuTIKqfvLDkk10UVKRdhI92g1wqgibR6CyXCNsEEEuE/Bxpi+9foa
oJQxXGIiDrHfP9hGzaQbPlPWCZfoMtFmvwcQMzF/Kele3KsvJiGP7j1XSxr7kg2t
GT8gjQluyvF8OB3E9PuAvIiOcQlLfTxMtPvGoZ46wEvysq+rujLO7tvxXRpcsqDo
HoBBDiTQsw3zdj5Y1gsCAwEAAaOCAgAwggH8MAwGA1UdEwEB/wQCMAAwDgYDVR0P
AQH/BAQDAgXgMB0GA1UdDgQWBBSu1tXYCxf/o3AKu9YioD9ftMFcPTCB6QYDVR0j
BIHhMIHeoYHYpIHVMIHSMQswCQYDVQQGEwJVUzERMA8GA1UECBMITmV3IFlvcmsx
DzANBgNVBAcTBlZpY3RvcjEiMCAGA1UEChMZUkVEQ09NIExhYm9yYXRvcmllcywg
SW5jLjEeMBwGA1UECxMVQ2VydGlmaWNhdGUgQXV0aG9yaXR5MTgwNgYDVQQDEy9S
RURDT00gTGFib3JhdG9yaWVzLCBJbmMuIENlcnRpZmljYXRlIEF1dGhvcml0eTEh
MB8GCSqGSIb3DQEJARYSc3VwcG9ydEByZWRjb20uY29tggEAMEoGA1UdHwRDMEEw
P6A9oDuGOWh0dHA6Ly9zb3kucmVkY29tLmNvbS9+c3dhbGwvcGhwa2kvaW5kZXgu
cGhwP3N0YWdlPWRsX2NybDCBhAYIKwYBBQUHAQEEeDB2MEYGCCsGAQUFBzAChjpo
dHRwOi8vc295LnJlZGNvbS5jb20vfnN3YWxsL3BocGtpL2luZGV4LnBocD9zdGFn
ZT1kbF9yb290MCwGCCsGAQUFBzABhiBodHRwOi8vbWljaGlnYW4ucmVkY29tLmNv
bTo4MDgwLzANBgkqhkiG9w0BAQUFAAOBgQBy75vP6SMJ4b9vXxjmry0FjtqiL+7b
vnuC251CzpqWZNmFiMv/GV/HrNonGX3DOo+3+T/2QgThOsmEs8DKNL8hPmoDRo3p
QLBRFMlAjKUEgYx6uNoh15Ec9G7D+q2NCIGCd2SqYex+EltFlVmDZMpacZlZeOXS
yQ8wq2ZFHkm+9g==
-----END CERTIFICATE-----

A1.3 TLS Ladder Diagram



A2 References

Forouzan, B. A. (2008). Cryptography and Network Security. New York, NY: McGraw Hill.

Trappe, W. and Washington, L. C. (2006). Introduction to Cryptography with Coding Theory. Upper Saddle River, NJ: Pearson Prentice Hall.