An Introduction to Public-Key Cryptography

By:

Matthew Sutton

May 5th, 2017

Advanced Writing for IS&T

Table of Contents

Figures & Tables:

Introduction

Public-key cryptography (asymmetric cryptography) is a cryptographic system for parties who wish to establish secure communications without the need to previously communicate a shared-secret. It has revolutionized digital communications by deploying the concept of *non-secret encryption* where a publicly facing element in the encryption process does not lead to compromise of confidentiality (Barr, 2002, p. 243). This allows entities who have never communicated to be able to start interacting securely at any time. Public-key cryptography is the main foundation for secure communication online.

Each entity using public-key cryptography has a mathematically intertwined key-pair consisting of an outward-facing *public-key* (the "non-secret" element of non-secret encryption), and a confidential *private-key*. To send a message using public-key cryptography, obtain the public-key of your recipient and use it to encrypt the message. Anything encrypted with your recipient's public-key can only be decrypted with the same recipient's corresponding private-key. If the private-key is properly secured, there is an extremely low risk of cracking data encrypted using public-key cryptography. Overall, public-key cryptography provides the following capabilities:

- ***Key Exchange*** – Public-key cryptography is most commonly used to exchange encryption keys for faster, stronger encryption algorithms.

- ***Authentication via Digital Signatures*** – By reversing the encryption process, an entity can "sign" a message via encryption with their private-key. This message is then can only be decrypted by using the same entity's public-key. Because the private-key is confidential, successful decryption of a digitally signed message authenticates an entity to be who they say they are.

- ***Sender Non-Repudiation*** – *Non-repudiation* is the inability to refute the origin of a message. Public-key cryptography provides non-repudiation via digital signatures. If an entity sends a digitally signed message, they are unable to refute the origin of the message because only they could have created the digitally signed message using their private-key.

- ***Message Authentication*** – A one-way *hashing algorithm* is used to generate a unique set of characters that is separately attached to a public-key message. After decryption of the message, a user compares hashes and determines if data integrity was lost. Obtaining the same hash authenticates the integrity of the received message.

This report is organized for readers at different levels of cryptography knowledge. For readers who have little to no cryptography knowledge, it is recommended to continue to the next page to begin at Cryptography Fundamentals. For readers who are familiar with cryptography, it is recommended to start on page 6 to review Secret-Key Cryptography.

Cryptography Fundamentals

What is cryptography?

Cryptography is defined as "the science and art of designing and using methods of message concealment" (Barr, 2002, p. 2). It is considered an art due to the finesse required to develop reversible, yet complex, mathematical functions also known as *ciphers*. Typically, ciphers apply a *key* in the mathematical function where the key must be kept secret or a message can be decoded with ease. Fundamentally, cryptography takes readable data, called *plaintext*, and applies a cipher to create a new form of incomprehensible data called *ciphertext*.

What is a cryptosystem?

Entities using cryptography apply two processes, entitled *encryption* and *decryption*, to complete a *cryptosystem*. Encryption is defined as the process of encoding plaintext using a cipher to create ciphertext. Decryption is defined as the process of decoding ciphertext to reveal the original plaintext. Essentially, cryptosystems are a series of reversible mathematical functions used in conjunction to achieve *confidentiality*. Confidentiality is defined as the confidence that there has been no unauthorized access to data. Confidentiality is one of the prime objectives in the information assurance world. Figure 1 shows a basic illustration of a cryptosystem and visually applies the terms encryption and decryption.

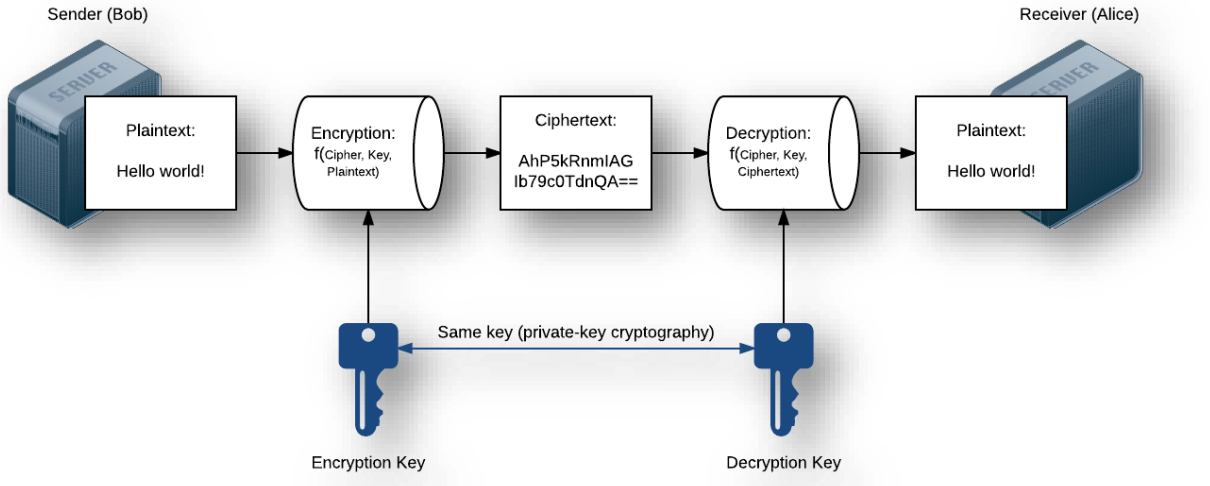


Fig. 1. Illustration of a Cryptosystem Implementing Secret-Key Cryptography

What are the different types of ciphers?

Substitution ciphers replace letters of the alphabet to encode the plaintext.

Plaintext	H	E	L	L	O	R	E	A	D	E	R
Encryption Substitution		E = X	L = Z	L = Z			E = X			E = X	
Ciphertext	H	X	Z	Z	O	R	X	A	D	X	R
Decryption Reverse Substitution	H	X = E	Z = L	Z = L	O	R	X = E	A	D	X =	R
Plaintext	H	E	L	L	O	R	E	A	D	E	R

Fig. 2. A Simple Substitution Cipher of Letter L with Z and E with X

Monoalphabetic ciphers are substitution ciphers where the cipher alphabet is fixed through the entire encryption process. The Caesar Cipher was a monoalphabetic shift cipher used in 50 B.C. (Barr, 2002, p. 5). It applied a constant shift substitution of the alphabet.

Cipher Alphabet:

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
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

Plaintext	H (7)	E (4)	L (11)	L (11)	O (14)	R (17)	E (4)	A (0)	D (3)	E (4)	R (17)
Encryption Shift +5	+ 5	+ 5	+ 5	+ 5	+ 5	+ 5	+ 5	+ 5	+ 5	+ 5	+ 5
Ciphertext	M (12)	J (9)	Q (16)	Q (16)	T (19)	W (22)	J (9)	F (5)	I (8)	J (9)	W (22)
Decryption Shift -5	- 5	- 5	- 5	- 5	- 5	- 5	- 5	- 5	- 5	- 5	- 5
Plaintext	H (7)	E (4)	L (11)	L (11)	O (14)	R (17)	E (4)	A (0)	D (3)	E (4)	R (17)

Fig. 3. Caesar Cipher with a Shift of 5

Polyalphabetic ciphers are substitution cipher where the cipher alphabet is changed during the encryption process. The Vigenere cipher shown on page 6 illustrates this example.

Transposition ciphers rearrange plaintext to create ciphertext. A *rail fence cipher* is a simple transposition cipher that uses two lines for transposition.

<table>
<tr><td>Plaintext</td><td>HELLO WORLD</td></tr>
<tr><td>Encryption
Shift +5</td><td>Place every other letter of plaintext in a different row.

<table><tr><td>Row 1</td><td>H</td><td></td><td>L</td><td></td><td>O</td><td></td><td>O</td><td></td><td>L</td><td></td></tr><tr><td>Row 2</td><td></td><td>E</td><td></td><td>L</td><td></td><td>W</td><td></td><td>R</td><td></td><td>D</td></tr></table>
Take Row 1 text and append Row 2 text to create the ciphertext.</td></tr>
<tr><td>Ciphertext</td><td>HLOOLELWRD</td></tr>
<tr><td>Decryption</td><td>Split cipher text in half and place intermittently into Rows 1 & 2.

HLOOL
ELWRD

<table><tr><td>Row 1</td><td>H</td><td></td><td>L</td><td></td><td>O</td><td></td><td>O</td><td></td><td>L</td><td></td></tr><tr><td>Row 2</td><td></td><td>E</td><td></td><td>L</td><td></td><td>W</td><td></td><td>R</td><td></td><td>D</td></tr></table>
Take weaving Row 1 followed by Row 2.</td></tr>
<tr><td>Plaintext</td><td>HELLO WORLD</td></tr>
</table>

Fig. 4. Rail Fence Cipher on plaintext HELLO WORLD

Why do we need cryptography?

> *Human beings have two inherent needs: (a) to communicate and share information and (b) to communicate selectively. These two needs gave rise to the art of coding the messages in such a way that only the intended people could have access to the information. Unauthorized people could not extract any information, even if the scrambled messages fell in their hand.*

Source: ("Origin of Cryptography", n.d., para. 1)

Cryptography is used to select who should have access to information. By using cryptography, only those who have the secret-key(s) will be able communicate, therefore, eliminating the middle-man threat. Most entities, whether it be business, individual, or government, require

their information to remain confidential. Credit card numbers, banking information, trade secrets, cryptographic keys, embarrassing Google searches, and classified information all require confidentiality due to the threat of malicious third-party. If confidential data is maliciously, or even inadvertently, obtained by a third-party they could use this private data to harm the data's proprietor. Damaging information can be leveraged for malicious purposes. Coercion, embarrassment, lost revenue, persistent spying, and loss of life are several of the possible outcomes of unauthorized disclosure.

We need cryptography because it attenuates the threat of unauthorized disclosure. Encrypted data that is exposed to third-party has no discernable information because the message is obfuscated. It cannot be used against its proprietor. Only those who have the secret-key(s) have access to the confidential information via decryption.

Can I trust that cryptography will secure my data?

Modern cryptography is designed to be strong enough to persist cracking attempts. The processing time required to brute-force a modern algorithm's decryption key ranges from decades to millennia. Modern algorithms are designed using *Kerckhoff's principle* of maintaining the assumption that the enemy will gain full familiarity with the cryptosystem (Krebs, 2015, para. 10). The algorithms and functionality of industry-standard cryptography are publicly accessible. Researchers and intelligence agencies around the world are continually working to break popular cryptosystems and apply security fixes.

Discussed further in the next section on secret-key cryptography, the *Advanced Encryption Standard* (AES) is the industry standard cryptosystem used in the United States. The cryptosystem for AES was selected by the National Institute of Standards and Technology (NIST) in 2000 during the Global Information Security Competition (Bulman, 2000). This international competition lasted for three years incorporating researchers from twelve different countries and "considerable private-sector cooperation" (Bulman, 2000, para 6). Researchers worked to attack over fifteen different algorithms to determine the best algorithm for the Federal Information Processing Standard. This goes to show how much effort is put into determining the strength of cryptosystems.

Ultimately, a user must secure their keys to maintain confidentiality. If a user does not maintain the safekeeping of their non-public encryption keys, they could become compromised. Proper key management must be exercised to maintain confidentiality.

Secret-Key Cryptography

Introduction

Secret-key cryptography is a form of cryptography where a same key is used for encrypting and decrypting.

Early History

"Secret-Key cryptography is the oldest form of encryption and has been used to safeguard communications for over three-thousand years" (Shelton, 2015). The first forms of cryptography belonged to the Ancient Egyptians. They would communicate by *hieroglyphs* where their language was only known to the scribes ("Origin of Cryptography", n.d., para. 6). Romans advanced cryptography from 400-500 B.C. by applying monoalphabetic ciphers such as in the Caesar Cipher ("Origin of Cryptography", n.d., para. 7). In 1553, polyalphabetic ciphers came to existence. The *Vigenere Cipher* applied a 26x26 grid where substitutions took place with cross-sectional lookups.

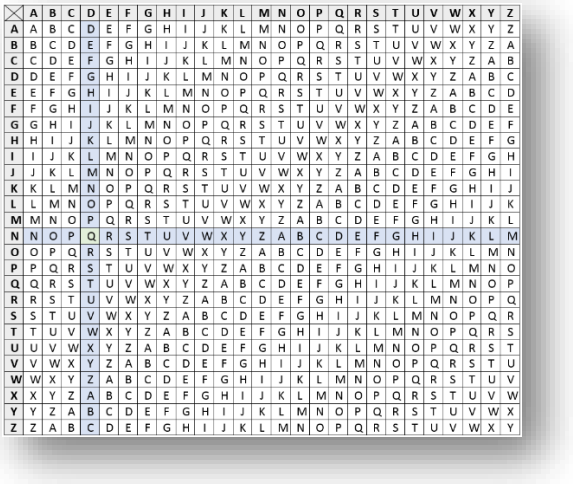


Fig. 5. A Vigenere Lookup of Letter N by key D to Reveal Ciphertext Q (Hammond, 2015)

Modern Secret-Key Cryptography

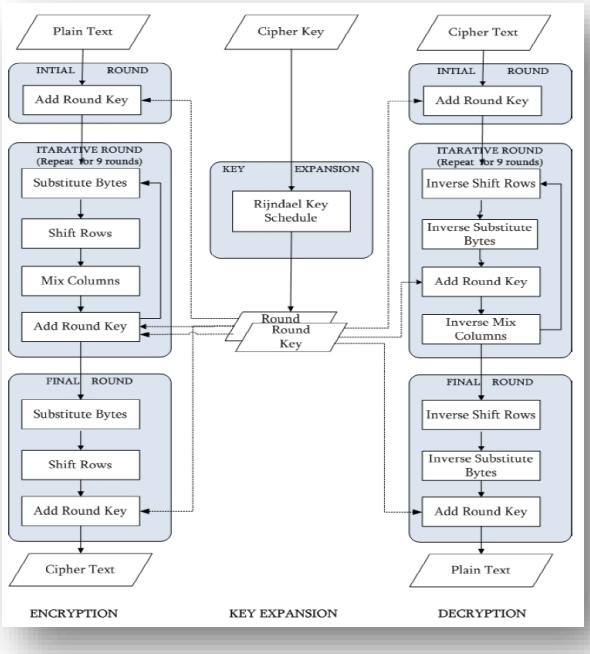
Modern cryptography (secret-key and public-key) heavily involves the usage of computers. Instead of operating on traditional characters, modern cryptography operates on *bits* ("Origin of Cryptography", n.d., para. 12). A bit is a 0 or 1. For example, the text you view on your PC is encoded using bits. A common encoding for text is the *American Standard Code for Information Interchange (ASCII)*. Each letter of the alphabet is associated with an integer number. Within ASCII, the lowercase letter 'a' is assigned decimal number 97. We can represent 'a' using bits by representing its associated decimal number in base 2 (binary) instead of base 10 (decimal).

$$97_{\text{base 10}} = 01100001_{\text{base 2}}$$

97 in Binary:	0	1	1	0	0	0	0	1	
Represented:	0×2^7	1×2^6	1×2^5	0×2^4	0×2^3	0×2^2	0×2^1	1×2^0	
	0 +	64 +	32 +	0 +	0 +	0 +	0 +	1 =	97

Table 1: Binary Representation of Lowercase ASCII Letter 'a'

The following AES information is not required for public-key comprehension, but it is included to show the complexity of modern-day cryptography. The AES cryptosystem operates on bits. Specifically, AES operates on 128-bit blocks of data at a time where the 128-bit block is organized as a 4x4 byte array called *State* (Hasib & Haque, 2008, p. 506). A *byte* is a grouping of 8 bits. The State array is modified over several rounds. The number of rounds is determined by the key size. AES' encryption and decryption uses four separate transformations during a single round:



- ***Substitute Bytes***: a non-linear substitution step where each byte is replaced with another byte according to a substitution table.
- ***Shift Rows***: a transposition step where each row of the state is shifted cyclically a certain number of steps.
- ***Mix Columns***: a mixing operation which operates on the columns of the state, combining the four bytes in each column. Applies matrix math.
- ***Add Round Key***: a bit wise exclusive OR operation is performed between each byte of the state and the round key which is generated from the cipher key using the Rijndael key schedule algorithm.

Fig. 6. AES 128-bit Algorithm Round Diagram and Accompanying Definitions (Hasib & Haque, 2008, p. 506)

What are the pros of secret-key cryptography?

Symmetric cryptography is designed to quickly encrypt and decrypt large amounts of data. The AES cipher is built into modern hardware. Intel processors have included AES instructions in their central processing unit (CPU) hardware since 2010 (Gueron, 2012, para. 1). This allows for "secure and high performance AES implementations" (Gueron, 2012, para. 5).

Modern forms of symmetric cryptography are exceptionally strong. Table 2 shows the possible number of keys for each AES key size form. It is theorized that not even the NSA can directly brute force AES encryption. Top security expert, Bruce Schneier (2012), states:

"…[My guess is the NSA doesn't] have a cryptanalytic attack against the AES algorithm that allows them to recover a key from known or

Key Size	Possible Key Combinations
128-bit	3.4×10^{38}
192-bit	6.2×10^{57}
256-bit	1.1×10^{77}

Table 2: AES Key Size to Possible Combinations (Arora, 2012, para. 5)

chosen ciphertext with a reasonable time and memory complexity."

To enumerate all possible AES 128-bit keys using a 2012 level supercomputer, it would take a billion-billion years (Arora, 2012, para. 14).

Key Size	Time to Enumerate all Keys
128-bit	1.02×10^{18} years
192-bit	1.872×10^{37} years
256-bit	3.31×10^{56} years

Table 3: 10.51 Petaflop Supercomputer Key Enumeration for AES (Arora, 2012, para. 14)

For 128-bit key size, there are approximately:

340,000,000,000,000,000,000,000,000,000,000,000,000 keys

This number is 340 followed by 36 zeros (Bulman, 2000, para 18).

What are the cons of secret-key cryptography?

The main downfall of secret-key cryptography is that there must be an alternative secure channel to transmit keys. "All secret key algorithms or systems require that the party generating the key share or transfer it to the other party in a secure manner" (Shelton, 2015). This is a major problem for first time communicators. They must be certain they exchange keys without an eavesdropper obtaining them.

A strictly secret-key encryption implementation for large scale networks is not feasible. Having a network of computers all communicating with each other using secret-key cryptography requires $n(n-1)/2$ keys (Odlyzko, 1994, p. 19). 20 million users all using symmetric cryptography with one-another would require 200 trillion total keys (Odlyzko, 1994, p. 19). This is not practical because of the large storage overhead to store trillions of keys.

How do we solve the flaws in secret-key cryptography?

Public-key cryptography is the answer to the flaws of secret-key cryptography. We use public-key cryptography for *key exchange*, *user/entity authentication*, *sender non-repudiation*, and *message authentication*. Key exchange solves the secure transmission problem. Public-key cryptography also has a significantly smaller storage overhead for communication. 20 million total users all communicating using public-key cryptography requires 40 million keys as opposed to 200 trillion keys.

Public-Key Cryptography

Introduction

Public-key cryptography is a form of cryptography that uses a pair of keys to encrypt and decrypt. This cryptosystem works by creating a public and private-key-pair for every party. Each party will make their *public-key* openly accessible, and will store their *private-key* for their eyes only. To send a message using public-key cryptography, a sender will encrypt their message using their recipient's public-key and then transmit the ciphertext to them. The recipient then decrypts the ciphertext with their private-key. This enables the concept of *non-secret encryption*.

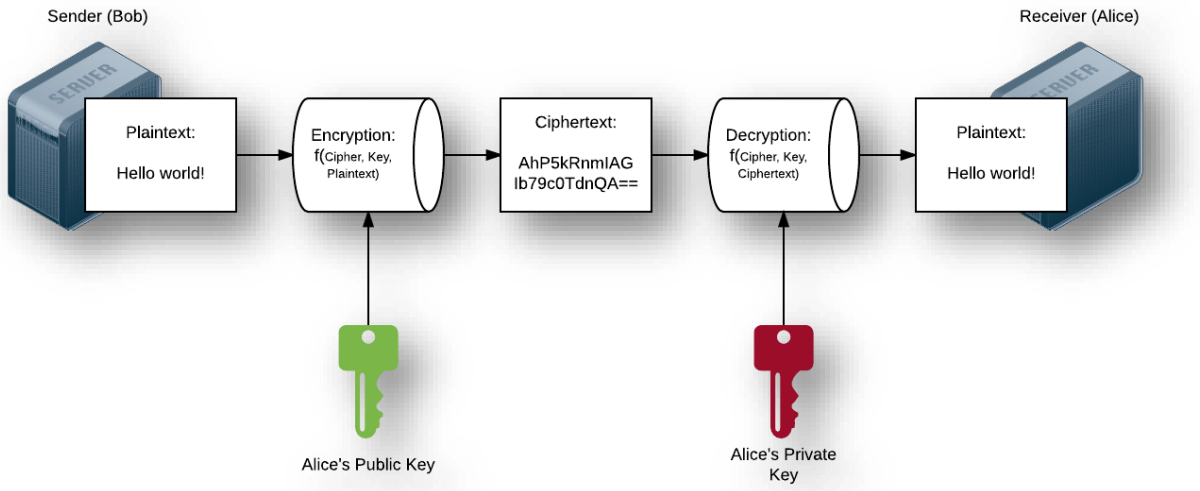


Fig. 7. Illustration of a Cryptosystem Implementing Public-Key Cryptography

History

For over three-thousand years there was no viable key exchange method for establishing secure communications (Shelton, 2015). Humans were stuck using secret-key cryptosystems (symmetric cryptography) in which parties need a separate, secure channel for key exchange. The process of securely communicating keys over unsecured mediums was not thought up until the 1960s. Government Communications Headquarters (GCHQ) cryptographer, James H. Ellis, was the first to write on the need for *non-secret encryption*. Non-secret encryption is defined as the concept of having a cryptosystem where some functions of the encryption process are publicly-facing without leading to compromise (Barr, 2002, p. 243). Without knowing about the GCHQ classified journal, 1970s Stanford Researchers Whit Diffie, Ralph Merkle, and Martin Hellman invented the first *public-key cryptosystem* and, therefore, solved the key exchange problem (Odlyzko, 1994, p. 19). Humans have only been using public-key cryptography for approximately half a century as compared to secret-key's multi-millennia existence.

What uses public-key cryptography?

Here are a few common applications of public-key cryptography *(Moulds, n.d.)*:

- Encrypted E-mail,
- Internet Traffic Encryption via Secure Sockets Layer/Transport Layer Security (HTTPS)
- E-Commerce
- Mobile Banking
- Authentication of Smart Devices
- Citizen Passports
- Mass Transit Ticketing Systems

What are the facets of public-key cryptography?

- ***Key Exchange***
  Public-key cryptography has enabled the ability for mass *key exchange* without the need to store the inexplicably large number of keys it would take for secret-key encryption to achieve this purpose. Key exchange is the most common use of public-key cryptography.

- ***Authentication via Digital Signatures***
  Authentication is defined as obtaining the identification of the sender (Shelton, 2015). Digital Signatures reverse the typical sending process: sign a message with your private-key and let others decrypt with your public-key. This means it's guaranteed to have come from you and creates non-repudiation.

- ***Sender Non-Repudiation***
  Non-repudiation is the inability to refute the origin of a message. Public-key cryptography provides non-repudiation via digital signatures. If an entity sends a digitally signed message, they are unable to refute the origin of the message because only they could have created the digitally signed message using their private-key.

- ***Message Authentication***
  A one-way *hashing algorithm* is used to generate a unique string of text that is separately attached to a public-key message. After decryption of the message, a user compares hashes and determines if data integrity was lost. Obtaining the same hash authenticates the integrity of the received message.

Public-Key Examples

The RSA Cryptosystem

The methodology for RSA requires the use of *prime numbers* and *modulus*, but is still secure today (Odlyzko, 1994, p. 19). A large key size needs to be used to guarantee longevity. Alternatively, changing keys after a certain timeframe works to ensure security. RSA relies on the *integer factorization problem* where computers have a hard time calculating the factors for large numbers. Today, computer systems can factor large primes with over 200 digits (Gaithuru, Bakhtiari, Salleh, & Muteb, 2015, p. 237). If a large number is created from two prime factors which are nearly the same size, no factorization algorithm is known which can solve the problem in reasonable time duration (Gaithuru, Bakhtiari, Salleh, & Muteb, 2015, p. 237). This is the principle behind RSA and why it is secure.

Public-key algorithms have lower performance; they run at a lower speed and have a higher memory requirement, but this is all compensated by their ability to be publicly displayed (Gaithuru, Bakhtiari, Salleh, & Muteb, 2015, p. 237). Here is a table comparing RSA to AES:

AES	**RSA equivalent bit length required for same level of security**
192-bit	7,680-bit
256-bit	15,360-bit

Table 4: AES to RSA Bit Security Level Equivalence (Gaithuru, Bakhtiari, Salleh, & Muteb, 2015, p. 237)

Therefore, it is common to use public-key cryptosystems to transmit keys for secret-key cryptosystems.

RSA Key Generation Algorithm

1. Choose two prime numbers denoted p and q.

2. Compute n:

$$n = pq$$

3. Compute φ (PHI):

$$\varphi = (p - 1)(q - 1)$$

4. Choose an e (public-key) such that:

   a. $1 < e < \varphi$

b. e and φ are *relatively prime*.

5. Use the *Extended Euclidian Algorithm* (not covered in this paper) to compute d (private-key) such that:

$$ed \equiv 1 \ (MOD \ \varphi)$$

Where $\equiv$ is the *congruence modulo* operator. This statement says: $e * d$ is congruent to $1 \ modulo \ \varphi$.

6. The public-key is (e, n). It may be broadcasted freely.

A sender encrypts their message by applying their recipient's public-key to following formula:

$$y = x^e \ MOD \ n$$

Where y is the resulting ciphertext and x is original plaintext.

After ciphertext calculations are complete, the sender transmits y to their recipient.

Note: x *can only range from* 0 *to* $n - 1$ *meaning small values for* p *and* q *only allow for small encrypted messages.*

7. The private-key is (d, n). It must be kept confidential.

Ciphertext y from Step 6 is decrypted by applying the private-key to the following formula:

$$x = y^d \ MOD \ n$$

RSA Key Exchange Example

Introduction

Note: This RSA example is heavily simplified. Small values for primes p and q are used to streamline RSA concepts. In modern RSA applications, RSA primes are 1024 to 2048 bits in length. This is beyond the audience's scope of learning. Additionally, Bob's example AES key has been shortened from a length of 128 bits to four characters (`EF1X`) for simplicity. AES keys are typically 128, 198, or 256-bits in length.

Bob wishes to transmit an AES key, `EF1X`, to Alice so they can engage in fast, secure data-transfer. Bob and Alice have not pre-shared any cryptographic keys, and are both connected to an unsecure network. They believe the network is compromised by eavesdroppers passively monitoring unencrypted traffic. Transmitting Bob's AES key to Alice in plaintext form could allow the eavesdropper to intercept it and use it against them. To mitigate this man-in-the-middle threat, Bob and Alice agree to use RSA to securely transfer Bob's AES key. See Figure 8 below for a diagram of the problem.

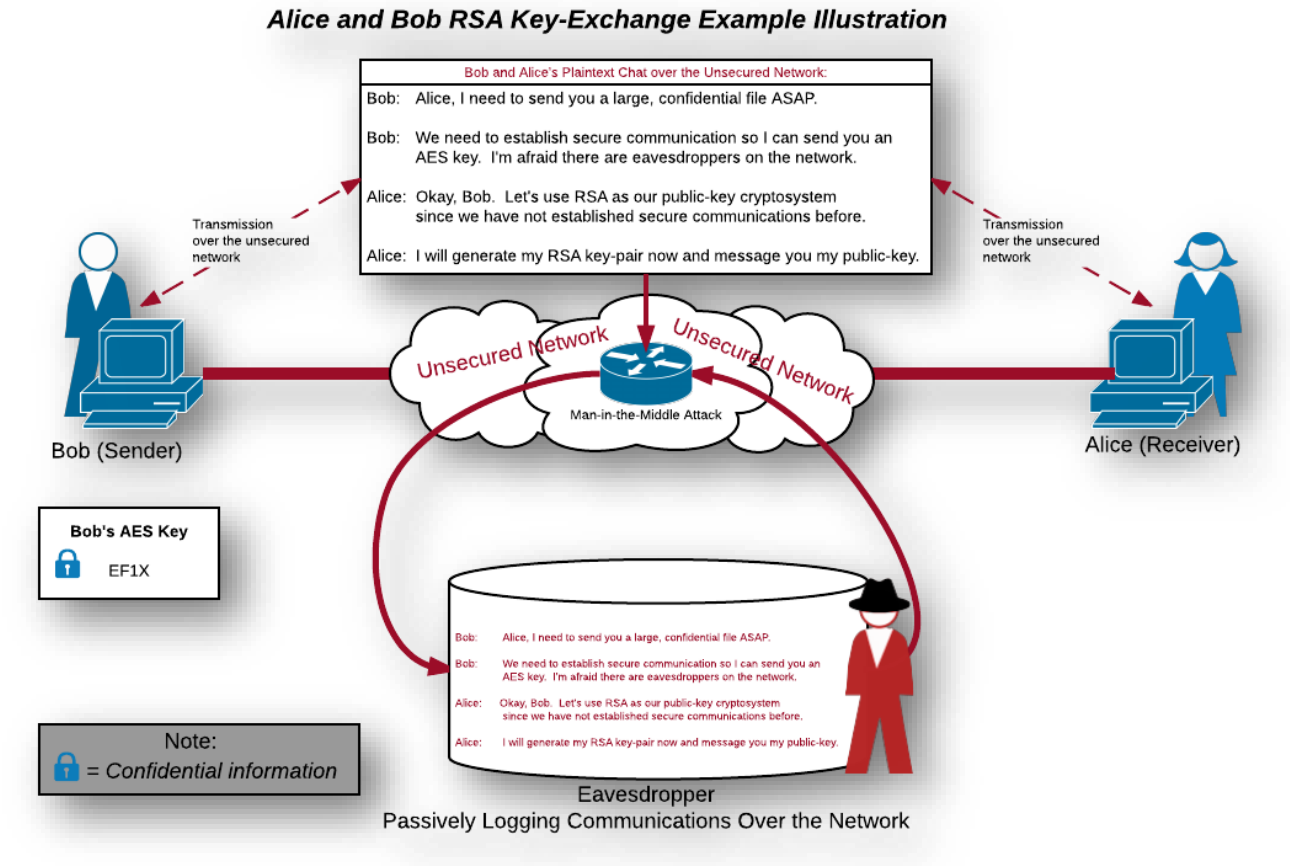


Fig. 8. Alice and Bob RSA Key Exchange Example Illustration

Generating Alice's RSA Key-Pair

Alice begins by selecting prime numbers $p = 3$ and $q = 11$. She then computes n and φ:

$$n = pq$$
$$n = 3 * 113$$
$$n = 33$$

$$\varphi = (p - 1)(q - 1)$$
$$\varphi = (3 - 1)(11 - 1)$$
$$\varphi = (2)(10) = 20$$
$$\varphi = 20$$

Alice must now select a public-key e such that $1 < e < \varphi$. Alice must also make sure that e and φ are relatively prime.

$$1 < e < \varphi$$
$$1 < 7 < 20 \text{ is true}$$
7 and 20 are relatively prime

Alice has chosen $e = 7$, so now she must use the Extended Euclidean Algorithm (not covered in this paper) to calculate the private-key d. This was calculated to be $d = 3$.

The numeric pair $(e, n) = (7, 33)$ is Alice's public-key, and her private-key is the numeric pair $(d, n) = (3, 33)$. Alice openly broadcasts her public-key to show she is enrolled in an RSA public-key cryptosystem and is ready to receive encrypted messages. Alice secures her private-key for her eyes only.

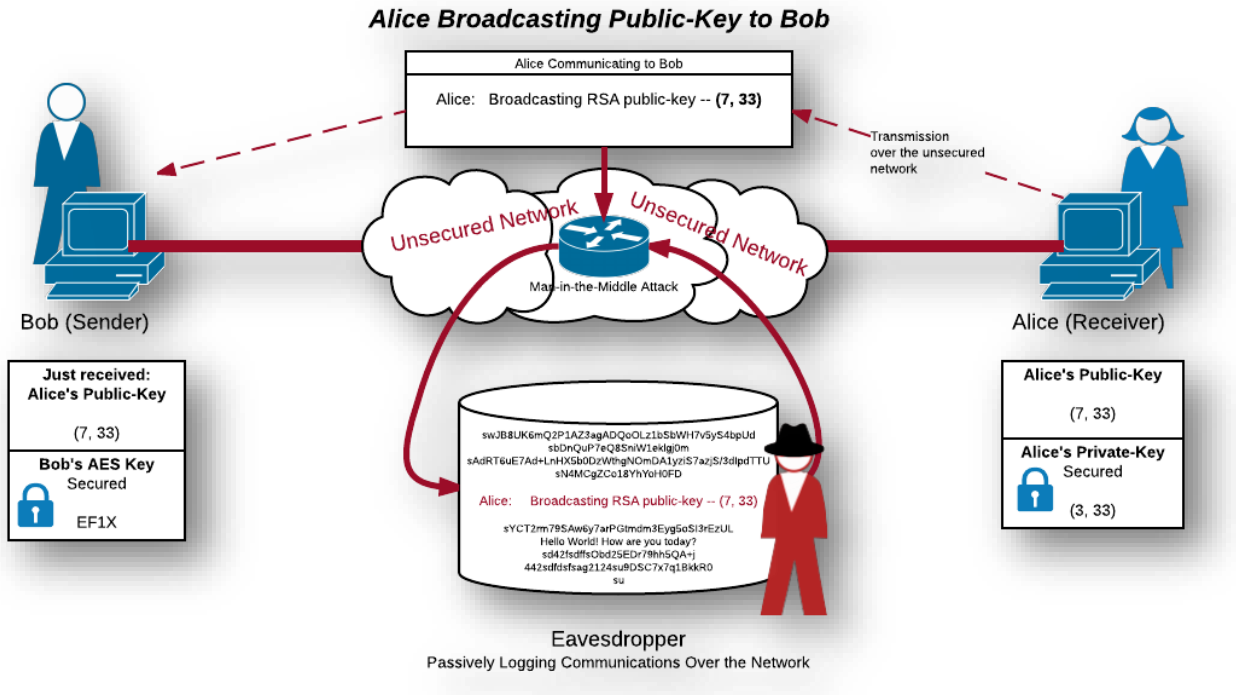


Fig. 9. Alice Broadcasting Public-Key to Bob

Encrypting Bob's AES Key Using Alice's Public-Key

Alice's broadcasted public-key, (7, 33), will be used to encrypt Bob's AES key. The AES key will be ran through the encryption equation $y = x^7 \; MOD \; 33$. Variable x is the AES key's plaintext letter that has been substituted with its associated number from the cipher alphabet (shown below in Table 5). For Bob's key EF1X, each plaintext letter will be calculated using its associated number:

Example Cipher Alphabet (0 to 32):																
Plaintext Letter	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
Associated Number	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Plaintext Letter	R	S	T	U	V	W	X	Y	Z	0	1	2	3	4	5	6	
Associated Number	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	

Table 5: Cipher Alphabet for RSA Key Exchange Example

Plaintext Letter to Associated number:

E = 4

F = 5

1 = 27

X = 23

Plugged into the encryption formula:

Encrypting E = 4

$y = 4^7 \; MOD \; 33$

$y = 16{,}384 \; MOD \; 33$

$y = 16$

Encrypting F = 5

$y = 5^7 \; MOD \; 33$

$y = 78{,}125 \; MOD \; 33$

$y = 14$

Encrypting 1 = 27

$y = 27^7 \; MOD \; 33$

$y = 10{,}460{,}353{,}203 \; MOD \; 33$

$y = 3$

Encrypting X = 23

$y = 23^7 \; MOD \; 33$

$y = 3{,}404{,}825{,}447 \; MOD \; 33$

$y = 23$

Therefore, computing $y = x^7 \; MOD \; 33$ for Bob's key `EF1X` yields ciphertext numbers: 16, 14, 3, and 23. Substituting 16, 14, 3, and 23 using Table 5's cipher alphabet yields the encrypted AES key `QODX`. It is now safe for Bob to transmit the encrypted AES key, `QODX`, to Alice over the unprotected medium. Only Alice's private-key *(3, 33)* can decrypt `QODX` back to its original form `EF1X`. As long as Alice keeps her private-key secure the eavesdropper cannot decrypt `QODX`.

Example Cipher Alphabet (0 to 32):																	
Plaintext Letter	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
Associated Number	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Plaintext Letter	R	S	T	U	V	W	X	Y	Z	0	1	2	3	4	5	6	
Associated Number	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	

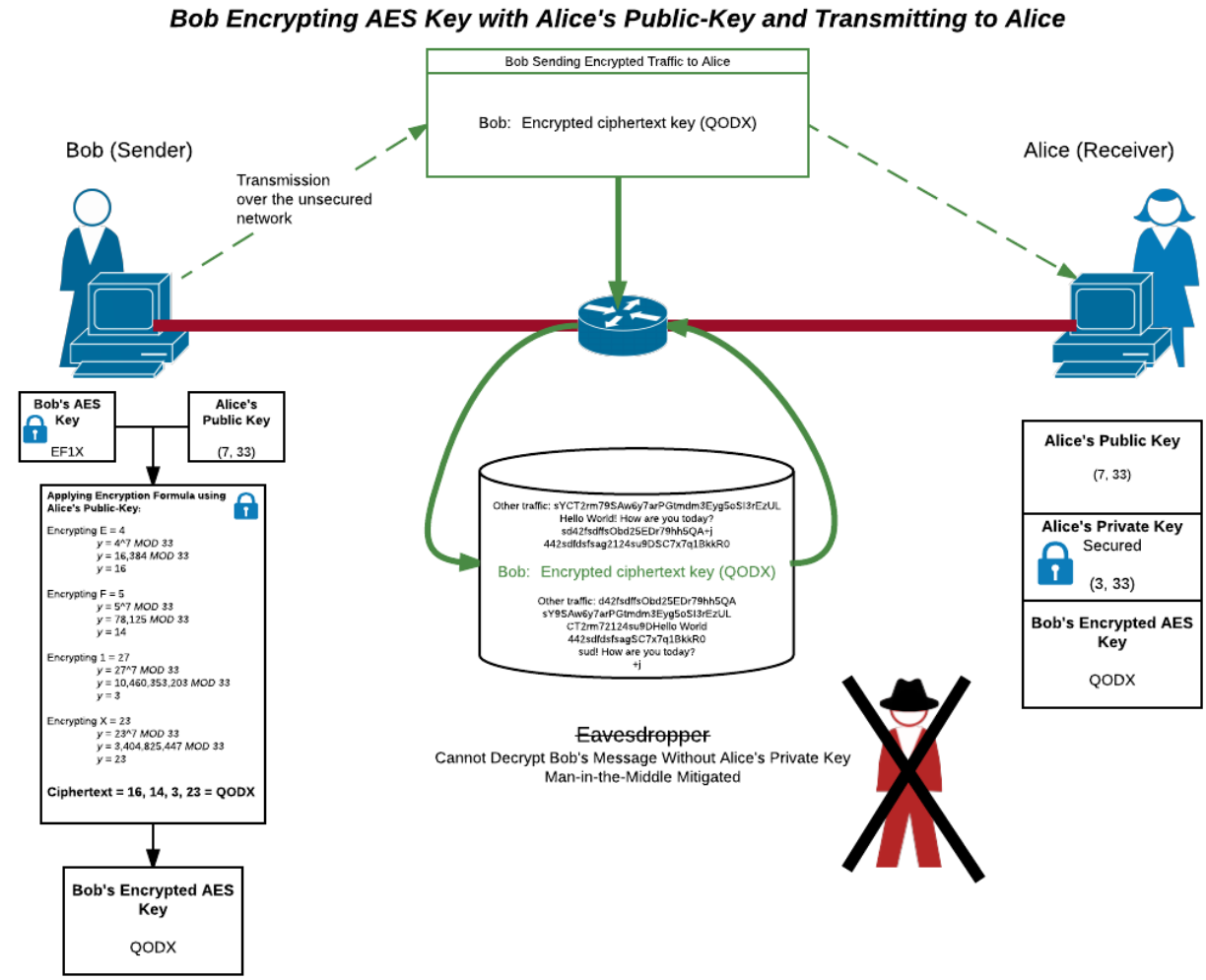


Fig. 10. Bob Encryption AES Key with Alice's Public-Key and Transmitting to Alice

Decrypting Bob's Encrypted AES Key Using Alice's Private-Key

Alice decrypts Bob's ciphertext using her private key (3, 33) where the decryption formula will be: $x = y^3 \; MOD \; 33$.

Decrypting Q = 16:

$x = 16^3 \; MOD \; 33$

$x = 4{,}096 \; MOD \; 33$

$x = 4$

$x = \mathrm{E}$

Decrypting O = 14:

$x = 14^3 \; MOD \; 33$

$x = 2{,}744 \; MOD \; 33$

$x = 5$

$x = \mathrm{F}$

Decrypting D = 3:

$x = 3^3 \; MOD \; 33$

$x = 27 \; MOD \; 33$

$x = 27$

$x = 1$

Decrypting X = 23:

$x = 23^3 \; MOD \; 33$

$x = 12{,}167 \; MOD \; 33$

$x = 23$

$x = \mathrm{X}$

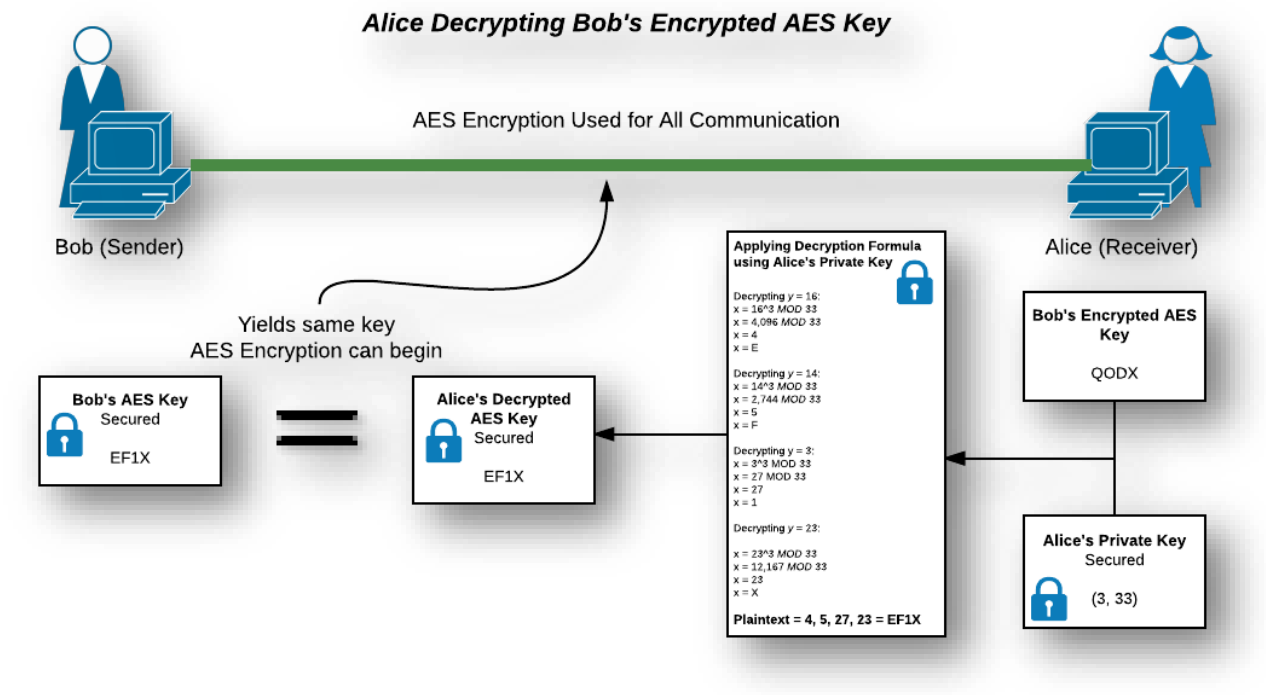


Fig. 11. Alice Decrypting Bob's AES Key

The decryption of QODX yields original plaintext numbers 4, 5, 27, and 23. Substituting the acquired plaintext numbers with the cipher alphabet returns Bob's original AES key EF1X.

Alice and Bob have successfully completed a RSA key exchange and can now begin to use AES for all their further communications. A more concise step-by-step breakdown of this example is included in the Appendix on the next page.

Conclusion

Public-key cryptosystems are used to enable non-secret encryption. By having a non-confidential public-key, users can broadcast a method to begin secure communications. If an entity protects their private-key's confidentiality, all the other entities using public-key cryptography can be reasonably sure that their communications are secure. Typically, public-key cryptography is used to engage in key exchange where another secret-key is transmitted. Secret-key cryptography is faster and more secure than public-key implementations. Without public-key cryptography, most communications on the Internet would be unsecure. It has become a ubiquitous part of everyday communication, and will continue to be for the rest of time.

Appendix

Step-by-Step Breakdown of RSA Key Exchange Example

Example Cipher Alphabet (0 to 32):																	
Plaintext Letter	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
Associated Number	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Plaintext Letter	R	S	T	U	V	W	X	Y	Z	0	1	2	3	4	5	6	
Associated Number	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	

Start both Alice and Bob at the same time:

Alice's Steps:

1. **Alice chooses prime numbers $p = 3$ and $q = 11$**
2. **Alice computes $n = (3)(11) = 33$**
3. **Alice computes $\varphi = 20$**
   $\varphi = (3 - 1)(11 - 1)$
   $\varphi = (2)(10) = 20$
   $\varphi = 20$
4. **Alice chooses her public-key $e = 7$**
   a. $1 < e < \varphi$
      $1 < 7 < 20$ is true
   b. 7 and 20 are relatively prime. The only positive factor they share is 1.
5. **Alice uses the Extended Euclidean Algorithm to calculate her private-key $d = 3$**
   $ed \equiv 1 \ (MOD\ \varphi)$
   $(7)(3) \equiv 1 \ (MOD\ 20)$
   $21 \equiv 1 \ (MOD\ 20)$
   $1 \equiv 1$
6. **Alice's public-key is now (7, 33).** Alice broadcasts this public-key to Bob.
   Encryption: $y = x^7 \ MOD\ 33$
7. **Alice's private-key is now (3, 33).** Alice must keep it confidential.
   Decryption: $x = y^3 \ MOD\ 33$

8 – 10. Alice waits for Bob's response

Bob's Steps:

1 – 7. Bob waits for Alice's public-key

8. **Bob receives Alice's public-key (7, 33).**
9. **Bob encrypts his AES key using Alice's public-key.**
   Bob's AES Key = EF1X
   Using the cipher alphabet shown above

   Encrypting E = 4
   $y = 4^7 \ MOD\ 33$
   $y = 16{,}384 \ MOD\ 33$
   $y = 16$

   Encrypting F = 5
   $y = 5^7 \ MOD\ 33$
   $y = 78{,}125 \ MOD\ 33$
   $y = 14$

   Encrypting 1 = 27
   $y = 27^7 \ MOD\ 33$
   $y = 10{,}460{,}353{,}203 \ MOD\ 33$
   $y = 3$

   Encrypting X = 23
   $y = 23^7 \ MOD\ 33$
   $y = 3{,}404{,}825{,}447 \ MOD\ 33$
   $y = 23$

10. **Bob transmits each piece of ciphertext to Alice.**
    $y = 16, 14, 3, 23$ = QODX

Alice's Final Steps:

11. **Alice receives Bob's ciphertext y = QODX = *16, 14, 3, 23***
12. **Alice decrypts Bob's ciphertext using her private-key (3, 33).**

Decrypting $y = 16$:	Decrypting $y = 14$:	Decrypting $y = 3$:	Decrypting $y = 23$:
$x = 16^3 \ MOD\ 33$	$x = 14^3 \ MOD\ 33$	$x = 3^3 \ MOD\ 33$	$x = 23^3 \ MOD\ 33$
$x = 4{,}096 \ MOD\ 33$	$x = 2{,}744 \ MOD\ 33$	$x = 27 \ MOD\ 33$	$x = 12{,}167 \ MOD\ 33$
$x = 4$	$x = 5$	$x = 27$	$x = 23$
$x =$ E	$x =$ F	$x = 1$	$x =$ X

13. **Alice has recovered Bob's AES key:** EF1X

Fig. 12. Step-by-Step Breakdown of RSA Key Exchange Algorithm

References

Arora, M., Sr. (2012, May 7). How secure is AES against brute force attacks? Retrieved March 26, 2017, from http://www.eetimes.com/document.asp?doc_id=1279619

Barr, T. H. (2002). *Invitation to Cryptology*. Upper Saddle River, NJ: Prentice Hall.

Bulman, P. (2000, October 02). Commerce Department Announces Winner of Global Information Security Competition. Retrieved April 26, 2017, from https://www.nist.gov/news-events/news/2000/10/commerce-department-announces-winner-global-information-security

Check our Numbers. (n.d.). Retrieved April 23, 2017, from https://www.digicert.com/TimeTravel/math.htm

Gaithuru J. N., Bakhtiari M., Salleh M., & Muteb A. M. (2015). A comprehensive literature review of asymmetric key cryptography algorithms for establishment of the existing gap. *2015 9th Malaysian Software Engineering Conference (MySEC)*, 236-244. doi: 10.1109/MySEC.2015.7475227

Gueron, S. (2012, August 2). Intel Advanced Encryption Standard (Intel AES) Instructions Set. Retrieved March 27, 2017, from https://software.intel.com/en-us/articles/intel-advanced-encryption-standard-aes-instructions-set

Hammond, J. (2015, April 24). Encryption 101: The Vigenère cipher. Retrieved March 21, 2017, from https://www.egress.com/en-US/blog/encryption-101-the-vigenere-cipher

Hasib, A. A., & Haque, A. A. M. M. (2008). A Comparative Study of the Performance and Security Issues of AES and RSA Cryptography. *2008 Third International Conference on Convergence and Hybrid Information Technology,2*, 505-510. doi:10.1109/iccit.2008.179

Krebs, B. (2015, August 18). How Not to Start an Encryption Company. Retrieved March 16, 2017, from https://krebsonsecurity.com/2015/08/how-not-to-start-an-encryption-company/

Moulds, R. (n.d.). What is a PKI and why is it important? Retrieved February 26, 2017, from https://www.thales-esecurity.com/blogs/2013/march/what-is-a-pki

Odlyzko, A. M. (1994). Public-key Cryptography. *AT&T Technical Journal, 73*(5), 17-23. doi:10.1002/j.1538-7305.1994.tb00606.x

Origin of Cryptography. (n.d.). Retrieved March 22, 2017, from https://www.tutorialspoint.com/cryptography/cryptography_quick_guide.htm

Schneier, B. (2012, March 22). Can the NSA Break AES? Retrieved March 26, 2017, from https://www.schneier.com/blog/archives/2012/03/can_the_nsa_bre.html

Shelton, B. K. (2015, November 11). Introduction to Cryptography. Retrieved February 26, 2017, from http://www.infosectoday.com/Articles/Intro_to_Cryptography/Introduction_Encryption_Algorithms.htm