PUBLIC KEY CRYPTOGRAPHY

MATTHEW LEUNG AND SATVIK BALAKRISHNAN

Abstract. This paper delves into the the world of public-key cryptography, providing an introductory exploration of its core concepts and essential definitions. Once the fundamentals are established, we delve into familiar public-key cryptographic examples, alongside examples we've developed. Shifting from theory to application, the paper will go over how public-key cryptography functions in the real world, assessing its speed, security, authentication, and energy consumption. This paper goes from foundational principles to practical implications, revealing the intricacies of cryptographic technologies.

Contents

Date: November 2023.

Introduction

Public-key cryptography is a core concept in modern cryptographic systems which allows you to securely share data. Through utilizing key pairs - each comprising a private and a public key - the private key remains confidential through a one-way mathematical function, while the public key is publicly shared. This allows anyone possessing the public key to encrypt a message securely, and only those with the private key can decipher it. More formally:

Definition 0.1. Public-key cryptography, a *key pair* consists of a *private key* and a *public key*. The private key is kept confidential, and the public key is openly shared for encryption and verification.

Public-key cryptography (also known as asymmetric cryptography) surpasses traditional methods in many aspects. It eliminates the need for an in-person meet up to exchange a key used to encipher and decipher data, thus decreasing the risk of unauthorized access. Since the public key is released and the private key is not, intercepting the public key does not give you access to the encrypted message.

Furthermore, public-key cryptography introduces the concept of digital signatures. A digital signature is a method to authenticate who the message you are receiving came from. Since only the public key can decipher a message encrypted by the private key, the recipient can be assured that they received the message from the correct person.

Definition 0.2. A *digital signature* in public-key cryptography is a unique cryptographic identifier created using a private key. It verifies the authenticity and integrity of a message when verified with the corresponding public key.

This layer of security enhances trust in digital interactions, particularly in scenarios where message integrity and sender verification are crucial. Thus, public-key cryptography allows for messages to be confidential and securely authenticated.

In this paper, we will start by introducing the basic terminology and examples of public key cryptography. After we have finished our introduction, we will move onto more advanced variations, and conclude with a comprehensive overview of practical applications.

1. Foundational Examples

Understanding some basic asymmetric cryptographies is crucial to developing our own and [do something to publish ourselves somewhere]. Let's take a look at two of the most common examples.

1.1. Rivest-Shamir Adleman.

The first example we will take a look at is *Rivest-Shamir Adleman* or RSA which utilizes modular exponentiation to create the asymmetric cryptography.

Definition 1.1. The remainder when an integer b is raised to the power e, and divided by positive integer m ; that is $c \equiv b^e \mod m$ is known as the *Modular exponentiation*.

The principle behind RSA is the fact that if we chose chose 3 large integers e, d, and n such that with modular exponentiation for all integers m with $0 \leq m < n$:

$$(m^e)^d \equiv m \mod n,$$

then even when knowing e and n, it can be very difficult to find d. In this case the *public key* is represented by n and e and the *private key* is represented by d. The message is represented by m.

Let's now generate the keys:

First, we chose two large prime numbers p and q. Both of these should be kept secret. We then use n as our modulus for both the public and private key. $n = pq$. We then compute $\lambda(n)$ where λ is Carmichael's Totient function.

Definition 1.2. The *Carmichael Totient Function* $\lambda(n)$ is the smallest positive integer m such that $a^m \equiv 1 \mod n$ hold for every integer a coprime to n.

Then we chose an integer e such that e and $\lambda(n)$ are coprime. Finally we let $d \equiv e^{-1} \mod \lambda(n)$. The private key would be d and the public key is e.

Practical Application. Suppose Carl wants to transfer a secret message with coordinates to a treasure map to Rebecca, then Rebecca shares her public key (n, e) to Carl. Carl then encrypts the coordinates and Rebecca can now decipher it with her code d that has not been shared.

- To send the coordinates m, Carl encrypts it by computing $m^e \mod n$ and sending the result K to Rebecca.
- To receive the coordinates m, Rebecca computes it by $K^d \equiv (m^e)^d \equiv m \mod n$.

You might be wondering how we would be able to send text messages via this encryption method. This is simple however due to the fact that the method of transferring data is secure. Therefore, we can take our message and use a code of some sort to convert it into numbers and send this. Once the recipient receives the number that you encoded, they can turn it into a normal text message by undoing the simple method you used to turn the text into numbers.

1.2. **Diffie-Hellman Key Exchange.** The second example we'll explore is the *Diffie-Hellman Key Exchange*, an encryption that allows two parties to establish a shared secret over an insecure channel. The key exchange relies on modular exponentiation, similar to RSA.

Let g be a primitive root modulo a prime number p. For a private key a, the remainder when g is raised to the power a and divided by p; that is $A \equiv g^a \mod p$, is known as the *Public Key* of party A.

The key idea of the Diffie-Hellman Key Exchange is in the ability to derive a shared secret despite transmitting the public keys over an insecure channel. If two parties, A and B, choose private keys a and b respectively, and exchange public keys A and B, then both parties can independently compute the shared secret:

$$(B^a) \equiv (A^b) \equiv g^{ab} \mod p$$

Even if an eavesdropper intercepts the public keys A and B, computing the shared secret without knowing either private key a or b is extremely difficult.

Key Generation. The key generation process involves the following steps:

- Choose a large prime number p and a primitive root g modulo p. These parameters are typically public.
- Each party, A and B, independently chooses a private key: a for A, and b for B.
- Compute the public keys: $A \equiv g^a \mod p$ for A, and $B \equiv g^b \mod p$ for B.

The public keys A and B are exchanged, and the shared secret is independently computed using the private keys.

Practical Application. Consider a scenario where Carl and Rebecca want to establish a secure method to transfer data:

- Carl and Rebecca agree on a prime number p and a primitive root g modulo p.
- Carl chooses a private key a and computes her public key $A \equiv g^a \mod p$.
- Rebecca chooses a private key b and computes his public key $B \equiv g^b \mod p$.
- They exchange public keys A and B over an insecure channel.
- Carl computes the shared secret $(B^a) \equiv g^{ab} \mod p$.
- Rebecca computes the shared secret $(A^b) \equiv g^{ab} \mod p$.

The shared secret is now known to both parties and can be used for secure communication.

A numerical version to show the computing power needed to decrypt a code like this is shown below:

Given $d = 997$ and $m = 1085323$ in the equation $D = f_z(x)^d \mod m$, we have:

$$
\begin{aligned}
f_{z_1}(x) &= 504602 &&\Rightarrow D_1 = (504602)^{997} \mod 1085323 = 3023 \\
f_{z_2}(x) &= 304785 &&\Rightarrow D_2 = (304785)^{997} \mod 1085323 = 1831 \\
f_{z_3}(x) &= 546341 &&\Rightarrow D_3 = (546341)^{997} \mod 1085323 = 1427 \\
f_{z_4}(x) &= 426932 &&\Rightarrow D_4 = (426932)^{997} \mod 1085323 = 2818 \\
f_{z_5}(x) &= 172949 &&\Rightarrow D_5 = (172949)^{997} \mod 1085323 = 2910 \\
f_{z_6}(x) &= 320683 &&\Rightarrow D_6 = (320683)^{997} \mod 1085323 = 2846 \\
f_{z_7}(x) &= 461096 &&\Rightarrow D_7 = (461096)^{997} \mod 1085323 = 1510 \\
f_{z_8}(x) &= 21952 &&\Rightarrow D_8 = (21952)^{997} \mod 1085323 = 2114 \\
f_{z_9}(x) &= 876194 &&\Rightarrow D_9 = (876194)^{997} \mod 1085323 = 2914 \\
f_{z_{10}}(x) &= 406630 &&\Rightarrow D_{10} = (406630)^{997} \mod 1085323 = 1710 \\
f_{z_{11}}(x) &= 261968 &&\Rightarrow D_{11} = (261968)^{997} \mod 1085323 = 2346
\end{aligned}
$$

The decrypted output is converted into a series of numerical values, which are then used to categorize various things. For instance, each pair of letters could correspond to an alphanumeric character.

2. Elliptic Curve Cryptography (ECC)

Elliptic Curve Cryptography is a powerful and efficient asymmetric cryptographic technique that leverages the properties of elliptic curves over finite fields. ECC is gaining popularity due to its ability to provide strong security with shorter key lengths compared to traditional methods like RSA.

Definition 2.1. An *Elliptic Curve* is defined by the equation $y^2 \equiv x^3 + ax + b \mod p$, where a, b, and p are parameters that define the curve. The curve is a set of points (x, y) that satisfy this equation, along with an additional point at infinity.

Elliptic Curve Cryptography is a powerful and efficient asymmetric cryptographic technique that leverages the properties of elliptic curves over finite fields. ECC is gaining popularity due to its ability to provide strong security with shorter key lengths compared to traditional methods like RSA.

Elliptic Curve Equation. The elliptic curve equation is given by:

$$y^2 \equiv x^3 + ax + b \mod p \tag{2.1}$$

Point Addition. If $P \neq Q$, the formulas for point addition are:

$$\begin{aligned} m &= \frac{y_Q - y_P}{x_Q - x_P} \mod p \\ x_R &= m^2 - x_P - x_Q \mod p \\ y_R &= m(x_P - x_R) - y_P \mod p \end{aligned}$$

If $P = Q$, the formulas are:

$$\begin{aligned} m &= \frac{3x_P^2 + a}{2y_P} \mod p \\ x_R &= m^2 - 2x_P \mod p \\ y_R &= m(x_P - x_R) - y_P \mod p \end{aligned}$$

Scalar Multiplication. The scalar multiplication algorithm is as follows:

Initialize $R =$ Point at Infinity
Express k in binary: $k = k_{\text{len}-1} \dots k_0$
For $i = \text{len} - 1$ to 0 :
  $R = 2R$
  If $k_i = 1$:
    $R = R + P$

Base Point Generation. The base point generation is given by:

$$G = k \cdot P \tag{2.2}$$

The security of ECC is based on the difficulty of the elliptic curve discrete logarithm problem. Given a point G on the elliptic curve and a public key $Q = kG$, it is computationally infeasible to determine the private key k.

Key Generation. The key generation process in ECC involves the following steps:

- Choose an elliptic curve and a base point G on the curve.
- Each party independently chooses a private key: k for Carl and l for Rebecca.
- Compute the public keys: $Q = kG$ for Carl and $R = lG$ for Rebecca.

The public keys Q and R are exchanged, and the shared secret is computed independently using the private keys.

Practical Application. Suppose Carl and Rebecca decide to employ Elliptic Curve Cryptography for secure communication:

- They agree on an elliptic curve and a base point G.
- Carl chooses a private key k and computes her public key $Q = kG$.
- Rebecca chooses a private key l and computes his public key $R = lG$.
- They exchange public keys Q and R over an insecure channel.
- Carl computes the shared secret $S = kR$.
- Rebecca computes the shared secret $T = lQ$.

The shared secrets S and T are now known to both parties and can be used for secure communication.

Security Advantages of ECC. One of the key advantages of ECC is its ability to provide equivalent security with much shorter key lengths compared to traditional algorithms. This is particularly important in resource-constrained environments, such as mobile devices or IoT devices, where shorter keys result in faster computations and lower energy consumption.

Moreover, ECC is resistant to attacks from both classical and quantum computers. The mathematical foundation of ECC makes it a robust choice for securing digital communications in the face of evolving technological threats.

Efficiency in Practice. In addition to its security advantages, ECC is known for its efficiency in practice. The computational requirements for key generation, encryption, and decryption are significantly lower compared to other asymmetric cryptographic methods. This makes ECC an attractive choice for applications where computational resources are limited.

2.1. **Example: Securing IoT Communication.** Consider a scenario where Carl has an Internet of Things (IoT) device that needs to securely communicate with Rebecca's server. Using Elliptic Curve Cryptography, they can establish a secure communication channel as follows:

- Carl's IoT device and Rebecca's server agree on an elliptic curve and a base point.
- The IoT device generates a private key and computes its public key.
- The device sends its public key to Rebecca's server.
- Rebecca's server, upon receiving the public key, computes the shared secret using its private key.
- Both parties now have a shared secret, which they can use for encrypting and decrypting messages exchanged between the IoT device and the server.

This example illustrates the practical application of Elliptic Curve Cryptography in securing communication between IoT devices and servers, where efficiency and strong security are paramount.

2.2. **Conclusion on ECC.** Overall, Elliptic Curve Cryptography offers a versatile and efficient solution for securing digital communication. Its mathematical foundation, based on the properties of elliptic curves, provides a high level of security with shorter key lengths. ECC's resistance to both classical and quantum attacks, along with its efficiency in computation, makes it an ideal choice for a wide range of applications, from secure messaging to IoT communication.

As we continue to advance in the digital age, the role of ECC in safeguarding sensitive information will become increasingly critical. Understanding and implementing Elliptic Curve Cryptography opens the door to a new era of secure and efficient communication in our interconnected world.

3. Comprehensive Overview of Public Key Cryptography Applications

3.1. **Enhancing Internet Communication Protocols.** Public key cryptography is a cornerstone in securing internet communication protocols, particularly in HTTPS-enabled web interactions. Protocols such as Transport Layer Security (TLS) and Secure Sockets Layer (SSL) employ this cryptography form to establish secure, encrypted connections between web clients and servers. This is achieved through a robust public key infrastructure (PKI), which facilitates the verification of identities and the encryption of data.

In these protocols, a key exchange mechanism is employed, often leveraging the mathematics of elliptic curves. Consider points P and Q on an elliptic curve and a scalar k. The process of establishing a shared secret, fundamental for secure communication, is mathematically expressed as:

$$\text{Shared Secret} = k \cdot P + k \cdot Q \tag{3.1}$$

This equation demonstrates the calculation of a shared secret that both the client and server can use to encrypt and decrypt the transmitted data, ensuring confidentiality and integrity.

3.2. **Blockchain Technology and Digital Signatures.** Blockchain technology, forming the backbone of cryptocurrencies like Bitcoin and Ethereum, heavily relies on public key cryptography. The integrity and authenticity of every transaction on the blockchain are maintained through digital signatures. These signatures, created using a combination of public and private keys, act as a robust mechanism to prevent fraud and unauthorized transaction alterations.

The digital signature process involves complex cryptographic algorithms. For a transaction to be validated, it must be accompanied by a signature that only the rightful owner of the digital asset can produce. The validation of a transaction, represented by a message m with a signature s and a public-private key pair (P, Q), is as follows:

$$\text{verify}(m, s, P) \rightarrow \text{True/False} \tag{3.2}$$

This verification ensures that each transaction is authentic and authorized by the holder of the private key, thereby maintaining the blockchain's integrity.

3.3. **Identity Authentication and Authorization in Digital Systems.** Public key cryptography is also pivotal in systems requiring secure identity authentication and authorization, such as Secure Shell (SSH) for remote server access and digital certificates for email encryption. The security provided by these systems hinges on the ability to reliably verify the identity of users or devices attempting to access sensitive resources.

The process typically involves a challenge-response mechanism. A server A presents a challenge to a user B, who must respond correctly using their private key. The server then verifies this response using the corresponding public key P, ensuring the user is legitimate. This process is succinctly captured in the following formula:

$$\text{verify}(P, \text{challenge}, \text{response}) \tag{3.3}$$

This method is widely used for secure logins, digital signing of documents, and encrypting emails, ensuring that only authorized parties can access or modify sensitive data.

3.4. **Securing Communications in Messaging Applications.** In the realm of modern messaging applications, such as Signal and WhatsApp, public key cryptography enables end-to-end encryption, ensuring that messages remain private between the sender and receiver. Each user in these systems possesses a unique pair of keys (P_A, Q_A) and (P_B, Q_B), used to encrypt and decrypt messages.

When user A sends a message m to user B, the message is encrypted using B's public key Q_B, resulting in a ciphered message c. This encryption ensures that only user B, with the corresponding private key, can decrypt and read the message:

$$c = \text{encrypt}(m, Q_B) \tag{3.4}$$

This encryption approach ensures the confidentiality and integrity of communications, safeguarding against eavesdropping and unauthorized access.

3.5. **Public Key Cryptography in IoT Security.** The burgeoning field of the Internet of Things (IoT) also benefits significantly from public key cryptography. In IoT ecosystems, numerous devices communicate and exchange data, often over unsecured networks. The implementation of public key cryptography in these environments ensures the confidentiality and integrity of the data transmitted between IoT devices and servers.

An IoT device D encrypts its data using its private key before transmitting it to a server S. The server, possessing the corresponding public key P_D, can decrypt and process the data securely. This process is crucial for maintaining the security of sensitive information in IoT applications:

$$\text{Encrypted Data} = \text{encrypt}(\text{data}, P_D) \tag{3.5}$$

The use of cryptography in IoT not only secures data transmission but also plays a vital role in authenticating devices, ensuring that only authorized devices can connect and interact within the network.

3.6. **Conclusion on Modern Applications.** In conclusion, public key cryptographies have evolved to become the cornerstone of modern digital security. From securing internet communication and blockchain transactions to enabling identity authentication and end-to-end encryption in messaging apps, the applications are diverse and widespread.

While the benefits of public key cryptographies in providing robust security are evident, challenges such as computational overhead, key management, and user-friendliness persist. Striking a balance between security and efficiency is an ongoing challenge in the dynamic landscape of digital communication.

As we continue to advance, the practical implementation of public key cryptographies will play a pivotal role in shaping the future of secure and trustworthy digital interactions. Understanding these applications not only enriches our theoretical knowledge but also equips us to navigate the complex realities of modern cryptographic systems.

References

[1] Dwi Liestyowati, *Public Key Cryptography. J. Phys.: Conf. Ser. (2020)*
[2] Whitefield Diffie, Martin, E. Hellman, *New Directions in Cryptography. IEEE. (1976)*