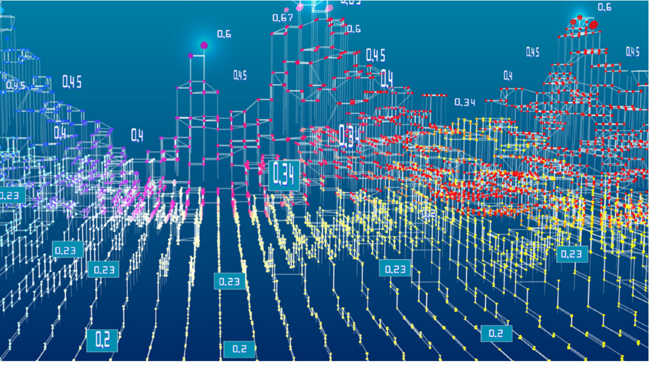


ElectronicDesign. Library

Cryptography: Why Do We Need It?

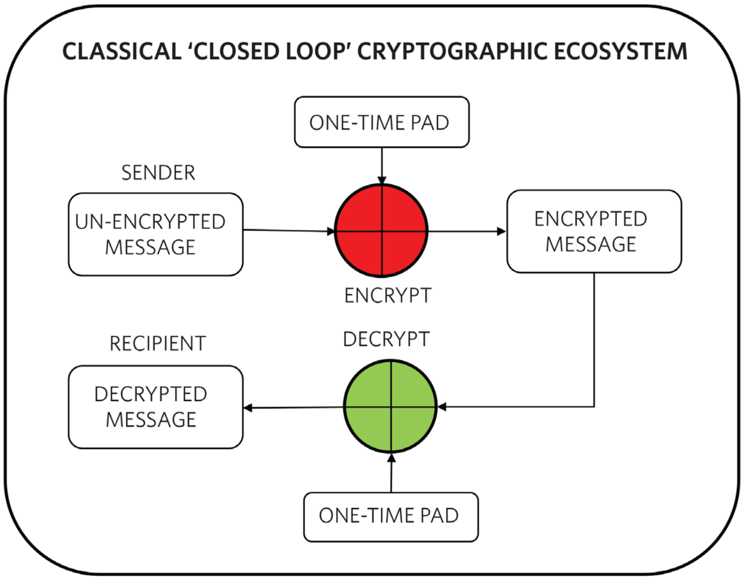
By ZIA A. SARDAR, Principal Member of Technical Staff, Maxim Integrated, now part of Analog Devices

This first article in "The Cryptography Handbook" addresses the escalating need for cryptography in this ever-more connected world.

In our day-to-day lives, the use of cryptography is everywhere. For example, we use it to securely send passwords over vast networks for online purchases. Bank servers and e-mail clients save your passwords using cryptography as well. Cryptography is used to secure all transmitted information in our IoT-connected world, to authenticate people and devices, and devices to other devices.

If all of the cryptographic engines/functions stopped working for a day, modern life as we know it would stop. Bank transactions wouldn't go through, internet traffic would come to a halt, and cell phones would no longer function. At this point, all of our important information would be exposed, and it then could be exploited to do unimaginable harm to us all.

Cryptography is an essential way of preventing that from happening. It secures information and communications using a set of rules that allows only those intended—and no one else—to receive the information to access and process it.



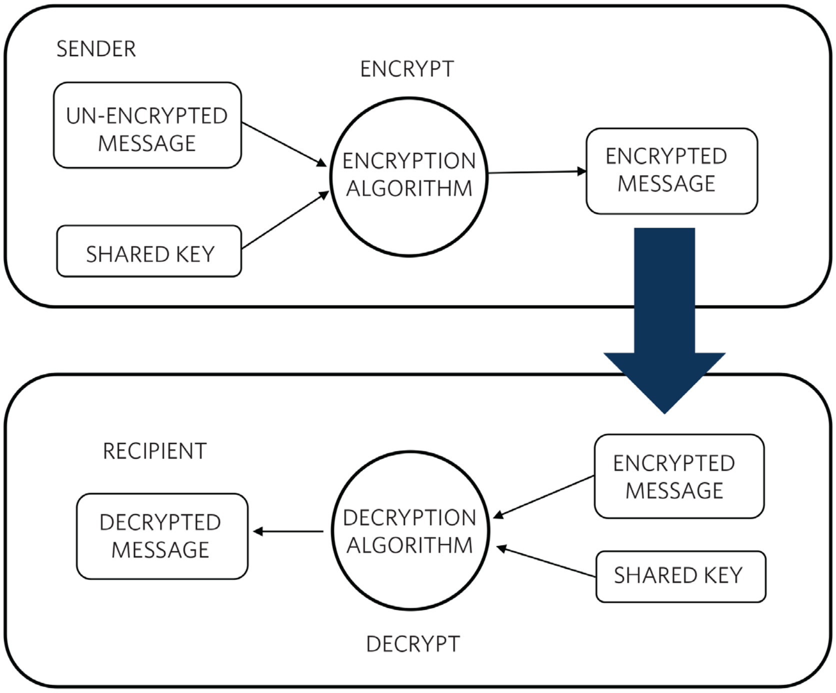
1. A classical closed-loop cryptographic system uses one-time pad as an encryption technique.

In the Old Days

Classically, cryptography used *"security by obscurity"* as way to keep the transmitted information secure. In those cases, the technique used was kept secret from all but a few, hence the term *"obscurity."* This made the communication secure, but it was not very easy to implement on a wide scale. Classical cryptographic methods are only secure when two parties can communicate in a secure ecosystem.

In **Figure 1**, we show a classical cryptographic system. The sender and the receiver first agree upon a set of pre-shared encryption/decryption keys. These keys are then used sequentially to encrypt and then de-crypt each subsequent message.

A MODERN SYMMETRIC KEY CRYPTOGRAPHIC SYSTEM



2. A modern symmetric key cryptographic system provides a greater level of security.

One-time pad is an encryption technique that requires the use of a pre-shared key that can be used only once. The same key must be used for encryption and decryption. The term *"One-Time Pad"* is an artifact from having each key on a page of a pad that was used and then destroyed. Once the pre-shared keys are exhausted, the sender and the receiver need to meet in a secure location to securely exchange a new set of keys and then store them in a secure location for the duration of the next set of message exchanges.

But...

Clearly, obsolete classical techniques are no longer viable. A vast system of electronic communication, commerce, and intellectual properties need to be secured across oceans and continents that would otherwise be intercepted by people with hostile intentions.

Next Phase of Cryptography

So, how do you implement an excellent level of security in such a massive system that can carry out billions of transactions in a short period? That's where modern cryptography comes in. It's an essential part of *secure but accessible* communication that's critical for our everyday life.

Next, we will learn how this is achieved on a day-to-day basis all around us. We rely on publicly known algorithms for securing the massive amount of information that's exchanged around the clock. These algorithms are standards-based and vetted in an open environment so that any vulnerabilities can be quickly found and addressed.

Figure 2 shows a simplified modern cryptographic system. Let's investigate these systems and algorithms a bit more in depth.

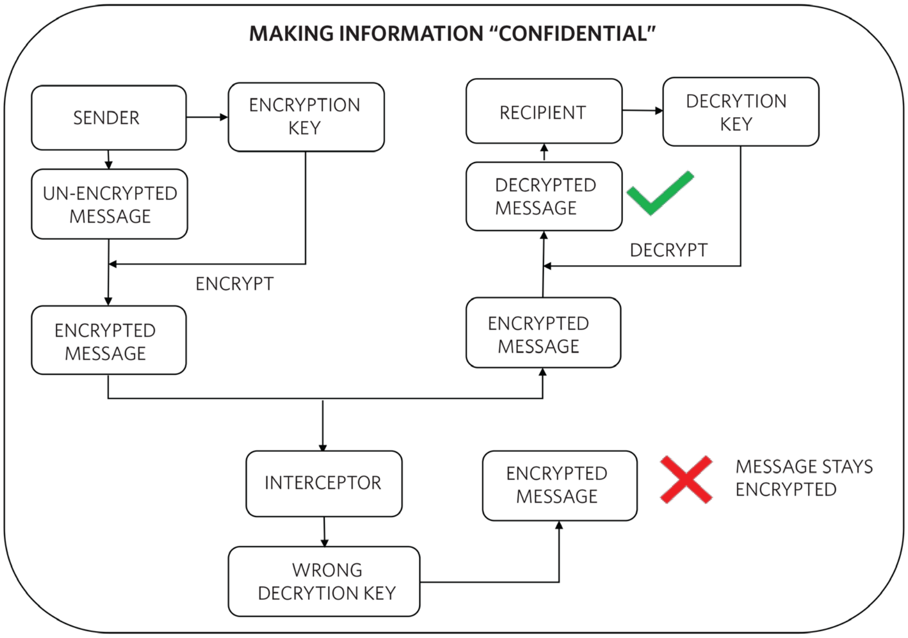
The basic tenet of a modern cryptographic system is that we no longer depend on the secrecy of the algorithm used, but rather rely on the secrecy of the keys. There are four essential goals of a modern cryptographic system:

- **Confidentiality:** Information can never be disclosed to someone who is not authorized to see it.
- **Identification and Authentication:** Before any information is exchanged, identify and then authorize both the sender and the recipient.
- **Integrity:** Information must not be modified in storage or transit. Any modification must be detectable.
- **Non-repudiation:** Cannot disclaim the creation/transmission of the message. This provides "digital" legitimacy and traceability of a transaction.

Current cryptographic systems provide all of the above or a combination of the above in various forms for an intended application. Let's explore each of these goals a little more to get a basic idea of how they are achieved.

Confidentiality

Confidentiality requires information to be secured from unauthorized access.



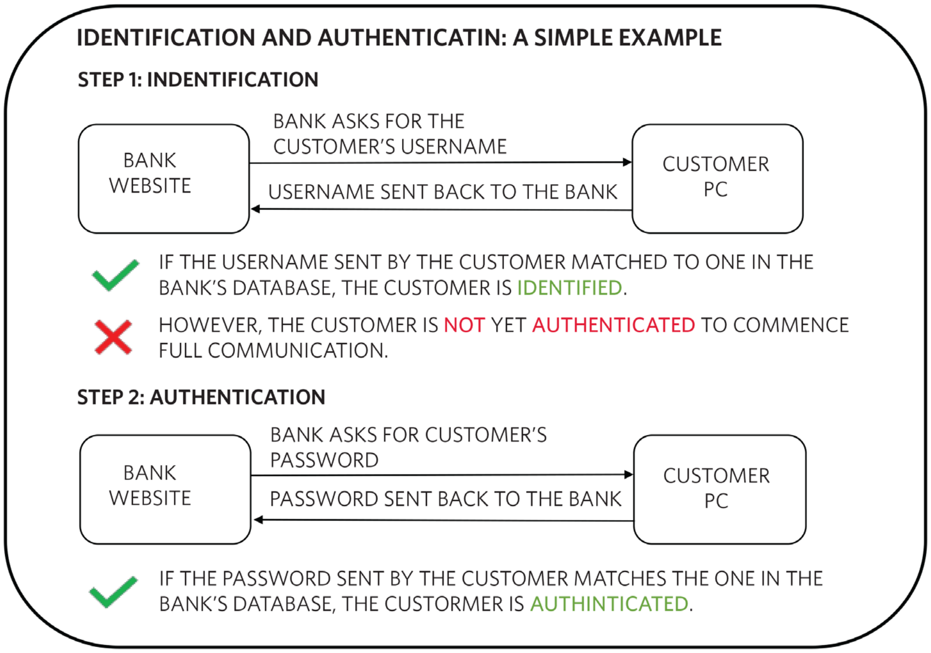
3. Encryption ensures information is kept confidential.

This is accomplished by encrypting a sent message using a cryptographic algorithm with a key that's only known by the sender and recipient. An interceptor might be able to obtain an encrypted message but will not be able to decipher it.

In **Figure 3**, we show how encryption is used. In this case, the sender and recipient have worked out a system to share the encryption/decryption key. They both use the key to encrypt/decrypt the messages they exchange between each other. If a malicious individual intercepts the message, no harm is done since that person will not have the key to decrypt the message. We'll discuss key sharing and encryption algorithms in a later article.

Identification and Authentication

The goal here is to first identify an object or a user and then authenticate them prior to initiating communication or other operations. Once the Sender has authenticated the Recipient, further communication can begin.


IDENTIFICATION AND AUTHENTICATIN: A SIMPLE EXAMPLE
STEP 1: INDENTIFICATION
BANK WEBSITE
BANK ASKS FOR THE CUSTOMER'S USERNAME
USERNAME SENT BACK TO THE BANK
CUSTOMER PC
✓ IF THE USERNAME SENT BY THE CUSTOMER MATCHED TO ONE IN THE BANK'S DATABASE, THE CUSTOMER IS IDENTIFIED.
✗ HOWEVER, THE CUSTOMER IS NOT YET AUTHENTICATED TO COMMENCE FULL COMMUNICATION.
STEP 2: AUTHENTICATION
BANK WEBSITE
BANK ASKS FOR CUSTOMER'S PASSWORD
PASSWORD SENT BACK TO THE BANK
CUSTOMER PC
✓ IF THE PASSWORD SENT BY THE CUSTOMER MATCHES THE ONE IN THE BANK'S DATABASE, THE CUSTORMER IS AUTHINTICATED.

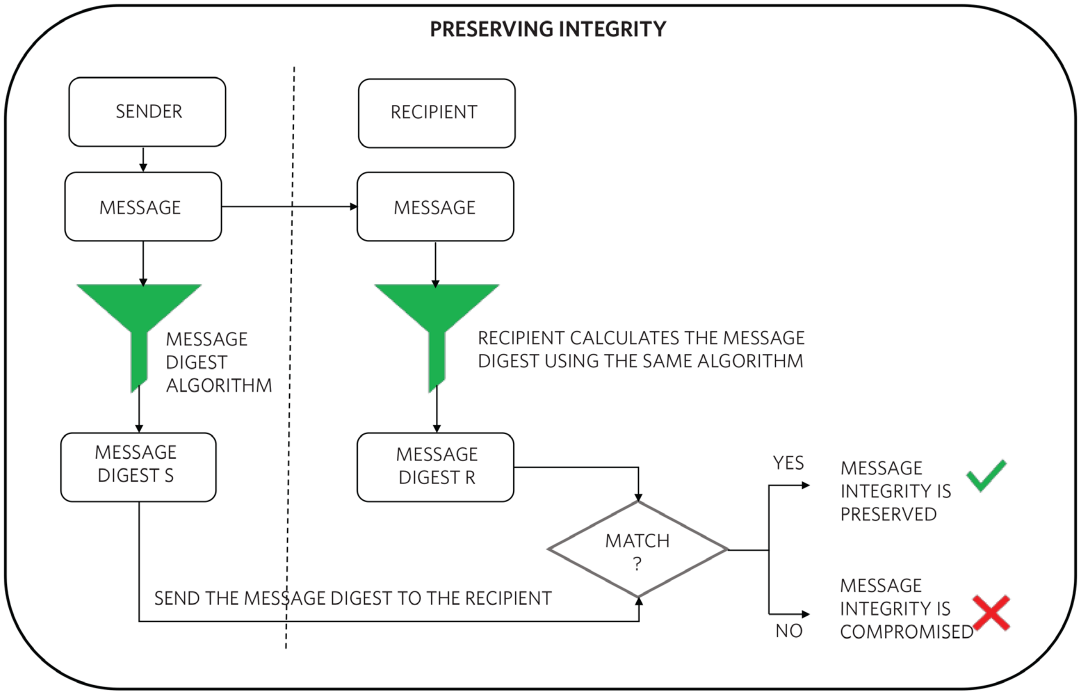
4. Identification and authentication, basic concepts of cryptography, shown in a simple example.

In **Figure 4**, we show how authentication works in one direction. The bank (Sender) authenticates the customer's PC (Recipient) using a simple username and password combination before letting the customer use the bank's website. The actual process is much more complex, but we are using this simple example to illustrate the basic concepts of cryptography. Identification and authentication can also be a bidirectional process, where the Sender and Recipient both need to identify each other before starting message exchanges. We will explore all those topics in detail later.

Integrity

How do we make sure that a message sent and then received over a communication network or data link hasn't been altered during transit? For example, there could be an attempt to intercept a message and insert a virus or malicious program to take control of the Recipient's PC or other equipment without their knowledge. To prevent this from happening, it's vital to ensure that any message transmitted isn't modified.

5. Using a message digest helps both a sender and recipient to preserve integrity.


PRESERVING INTEGRITY
SENDER
RECIPIENT
MESSAGE
MESSAGE
MESSAGE DIGEST ALGORITHM
RECIPIENT CALCULATES THE MESSAGE DIGEST USING THE SAME ALGORITHM
MESSAGE DIGEST S
MESSAGE DIGEST R
MATCH ?
YES
NO
MESSAGE INTEGRITY IS PRESERVED ✓
MESSAGE INTEGRITY IS COMPROMISED ✗
SEND THE MESSAGE DIGEST TO THE RECIPIENT

As shown in **Figure 5**, one way to do this is to use a message digest. The Sender and Recipient use an agreed-upon Message Digesting Algorithm to create and verify the message digest output. If the message is altered, the message digests will not match, and the Recipient knows that either tampering has occurred or there was a transmission error. Many Message Digesting Algorithms are used in modern cryptographic applications, including SHA-2 and most recently SHA-3. We will dive into the details in an upcoming *Cryptographic Algorithms* article.

Non-Repudiation

In a communication system where a multitude of messages are exchanged, there's a need to trace the incoming message back to the Sender. This is required to ensure that the Sender doesn't deny sending the message. Very similar to a pen-and-paper legal document where we sign on the dotted line to finalize a contract, a digital signature is used to achieve similar goals in the digital domain.

6. The non-repudiation process includes digital signature generation, transmission, C and verification.

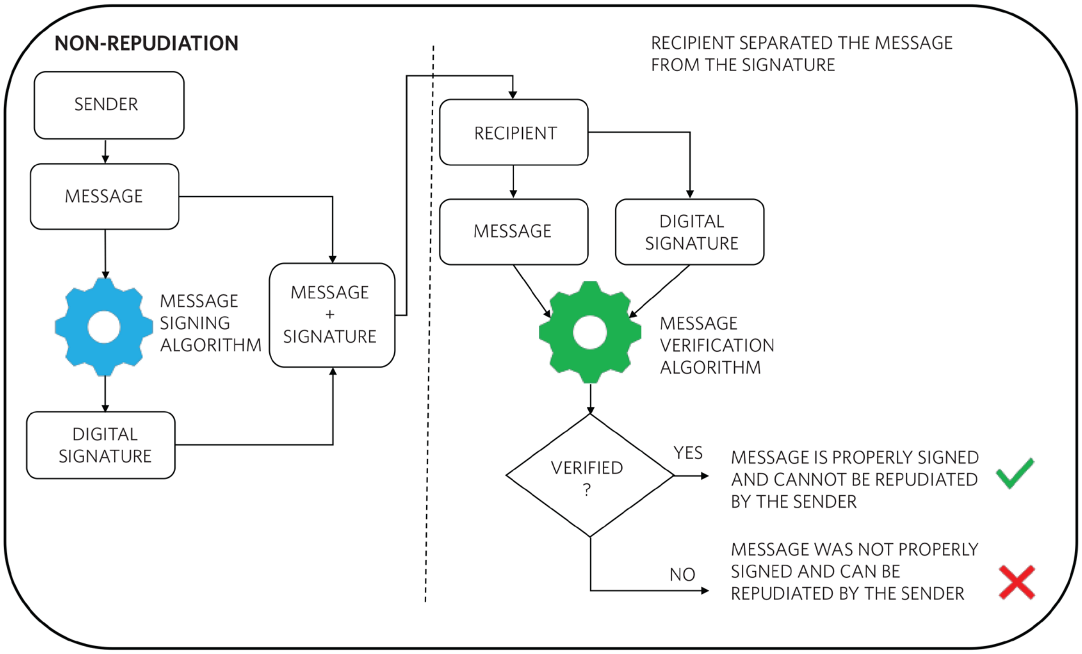

NON-REPUDIATION
SENDER
MESSAGE
MESSAGE SIGNING ALGORITHM
MESSAGE + SIGNATURE
DIGITAL SIGNATURE
RECIPIENT SEPARATED THE MESSAGE FROM THE SIGNATURE
RECIPIENT
MESSAGE
DIGITAL SIGNATURE
MESSAGE VERIFICATION ALGORITHM
VERIFIED ?
YES
MESSAGE IS PROPERLY SIGNED AND CANNOT BE REPUDIATED BY THE SENDER
✓
NO
MESSAGE WAS NOT PROPERLY SIGNED AND CAN BE REPUDIATED BY THE SENDER
✗

Figure 6 shows a simplified view of the digital signature generation, transmission, and verification process. First, the Sender takes the outgoing message and puts it through a Message Signing Algorithm to generate a digital signature related to the message and the Sender's verified identity. The Sender then attaches the digital signature to the original message and sends it to the Recipient. The Recipient takes the incoming combined message and separates the original message and the digital signature. Both are then input into a Message Verification Algorithm. The result can then be used by the Recipient to prove that the message was signed by the Sender. We will discuss digital signature algorithms in detail in the upcoming article on cryptographic algorithms.

ZIA SARDAR is an Applications Engineer at Maxim Integrated. Prior to joining Maxim, Zia worked at Advanced Micro Devices and ATI Technologies, focusing on graphics processors and PCIe bridge products. He holds an M.S. degree in computer engineering and a B.S. degree in electrical engineering from Northeastern University in Boston, Massachusetts.

to view this article online, ☞ *click here*