CYBERSECURITY
RESEARCH LAB

# CRL 101 Series

## Standardized Cryptographic Algorithms

By Annegret Henninger & Dr. Atefeh Mashatan

Cryptography is commonly used to protect digital information, to ensure its confidentiality, integrity, and authenticity. There are three kinds of cryptographic methods which can be used individually, and in combination:

- *Symmetric-key cryptography* uses the same, secret key to encrypt and decrypt;
- *Asymmetric-key cryptography* uses a public key to encrypt and a private key to decrypt; and
- *Hash algorithms* irreversibly convert a message into a fixed-length hash value.

Encryption is a cryptographic primitive whereby a plain-text message is passed through an encoding algorithm along with an established input called a 'key'. Symmetric and asymmetric-key cryptography are used to encode and subsequently decode messages to protect the confidentiality of the data. The illegible output is called ciphertext. Symmetric-key methods are typically more efficient than asymmetric methods achieving the same level of security. Hence, the most common approach, is to use a combination of both symmetric and asymmetric-key encryption methods, where an asymmetric-key encryption technique is used to share an established key between both parties, which is then used in symmetric-key encryption for subsequent communication.

Within symmetric-key cryptography, there are two categories of encryption algorithms: *block ciphers* and *stream ciphers*. In a block cipher, the message is broken down into fixed-length blocks before being encrypted one block at a time using the same key. In a stream cipher, on the other hand, the input message is encrypted one element at a time producing a stream of output. A stream cipher does not reuse an encryption key, but uses a keystream to generate random or pseudorandom keys which are used together with the algorithm to encrypt the inputs. While a stream cipher is typically faster and uses less data, block ciphers can host additional data integrity and authentication capabilities, and remain more common.

With asymmetric-key cryptography, each party has a key which is kept in a public register or available file, and a private key which remains secret. The encryption can be done in two ways to achieve either confidentiality or integrity. To ensure confidentiality, the sender uses the receiver's public key to encrypt the message and the receiver then uses their own private key to decrypt the message. This keeps the message confidential, as only the receiver should possess the private key needed to decrypt the message. Alternatively to achieve authentication and/or data integrity, the sender encrypts the message with their own private key, and the receiver decrypts the message with the sender's public key. Without the sender's private key, the message can't be tampered

with and re-encrypted to look authentic, thus protecting the authenticity and integrity of the data. However, this does not ensure confidentiality, as an intruder can decrypt the message with the sender's public key.

The security of any encryption scheme, whether symmetric or asymmetric, is dependent on the security of the algorithm itself and the encryption keys. Cryptanalysis tries to determine the security of the algorithm by determining the minimum amount of computational work needed to break a cipher. The length of the key is another security parameter and it needs to be long enough so that the space of all keys cannot be brute-forced in any reasonable time from an attacker's perspective.

Hash functions complement encryption algorithms. The purpose of a hash function is to protect data integrity and authentication. Similar to an encryption scheme, a hash function takes an input and outputs illegible content, but the output is always of fixed length. The hashing algorithm is created so that it is extremely difficult to find different data that will generate the same hash values. This makes the hash value act like a unique fingerprint, ensuring the authenticity and integrity of the data.

To protect data, the three cryptographic methods can be employed in combination. A hash algorithm can be added to an asymmetric-key exchange to ensure the authenticity and integrity of the data in the exchange. Then, a message can be encrypted using the established keys, and sent along with a hashed output of the encrypted message. The encryption protects the confidentiality, and the hash function ensures the integrity and authenticity of the data.

To ensure their security, cryptographic algorithms need to be extensively tested before they are widely accepted to be used. The algorithms need to be secure against current computers, and must also be able to protect data against the more powerful computers of the foreseeable future. To achieve this level of confidence about any of the cryptographic primitives, they must typically stand the test of time and due diligence where researchers from both academia and the industry examine them very closely trying every cryptanalytic tool in their toolbox. Some proposals fail to achieve the desired level of security in this process and some prevail to be good candidates for *standardized* cryptographic schemes. The aforementioned process is formalized by standardization bodies such as the National Institute of Standards and Technology (NIST) and its European counterpart, the European Telecommunications Standard Institute (ETSI), have the mandate to engage the cryptographic community before recommending any algorithm for commercial implementation.

CYBERSECURITY
RESEARCH LAB

NIST is currently recommending the following algorithms as its Commercial National Security Algorithm Suite:

| Algorithm | Parameters | Function |
|---|---|---|
| *Symmetric Cryptography* | | |
| Advanced Encryption Standard (AES) | 256 bit key | Uses a symmetric block cipher to protect information |
| *Asymmetric Cryptography* | | |
| Rivest, Shamir, and Adelman (RSA) | Minimum 3072-bit modulus | Uses an asymmetric algorithm for key establishment |
| Elliptic Curve Diffie-Hellman (ECDH) Key Exchange | Curve P-384 | Uses an asymmetric algorithm for key establishment |
| Diffie-Hellman (DH) Key Exchange | Minimum 3072-bit modulus | Asymmetric algorithm used for key establishment |
| Rivest, Shamir, and Adelman (RSA) | Minimum 3072-bit modulus | Uses an asymmetric algorithm for digital signatures |
| Elliptic Curve Digital Signature Algorithm (ECDSA) | Curve P-382 | Uses an asymmetric algorithm for digital signatures |
| *Hash* | | |
| Secure Hash Algorithm 2 (SHA 2) | SHA-384 | Uses SHA algorithm to compute a compressed representation of information |

Currently, new algorithms are being tested to protect information against quantum computing attacks. As standardization can be a long process, the CRL is developing classical/quantum hybrid solutions for present use, which protect against quantum attacks, while maintain current security guarantees.

July 2, 2019
CRL 101 series

The CRL 101 series is a knowledge transfer project designed to give readers an overview of trending security-related technologies that we are working on at the Cybersecurity Research Lab.