SHA1 and HMAC-SHA1 Implementation Documentation

**Overview**

This module provides Python implementations of the SHA-1 hashing algorithm and the HMAC (Hash-Based Message Authentication Code) using SHA-1 for message authentication and integrity check.

**Functions**

left\_rotate(value, shift)

Parameters:

value: int - The integer value to be rotated.

shift: int - The number of bits to rotate left.

Returns:

int - Rotated integer value.

Description:

Performs a bitwise left rotation on a 32-bit integer.

sha1(message)

Parameters:

message: bytes - The input message for SHA-1 hashing.

Returns:

bytes - The 160-bit (20-byte) SHA-1 hash value.

Description:

Computes the SHA-1 hash of an input message following the SHA-1 algorithm, processing it in 512-bit blocks and producing a 160-bit hash value.

hmac\_sha1(key, message)

Parameters:

key: bytes - The secret key for HMAC computation.

message: bytes - The message to be authenticated.

Returns:

bytes - The 160-bit HMAC value computed using SHA-1.

**Description:**

Computes the HMAC using SHA-1 as the underlying hash function. The key is utilized to generate two derived keys (inner and outer padding). These keys are then used to compute the inner and outer hash functions in the HMAC computation.

**What I have Learn**

Through using and understanding this module, I learn the following:

SHA-1 Algorithm: How SHA-1 processes messages and produces a 160-bit hash.

HMAC Mechanism: How HMAC provides message authenticity and integrity using an underlying hash function (SHA-1 in this instance) and a secret key.

**Error Handling**

I did not meet some big issue when coding, but the hmac.py took me long time to build it.

**Test result**图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成文本

描述已自动生成