Cryptography and Network Security

MAC

HMAC

CMAC

Session Meta Data

Author	**Dr T Sree Sharmila**
Reviewer	
Version Number	1.0
Release Date	19 July 2018

Revision History

Revision Date	Details	Version no.
		1.0

Agenda

- Introduction
  - Message authentication requirements
  - Message authentication using encryption
  - MACs
  - HMAC authentication using a hash function
  - CMAC authentication using a block cipher
- Summary
- Test your understanding
- References

Introduction

- message authentication is concerned with:
  - protecting the integrity of a message
  - validating identity of originator
  - non-repudiation of origin (dispute resolution)
- will consider the security requirements
- then three alternative functions used:
  - hash function
  - message encryption
  - message authentication code (MAC)

Agenda

- Introduction
  - Message authentication requirements
  - Message authentication using encryption
  - MACs
  - HMAC authentication using a hash function
  - CMAC authentication using a block cipher
- Summary
- Test your understanding
- References

Message Security Requirements

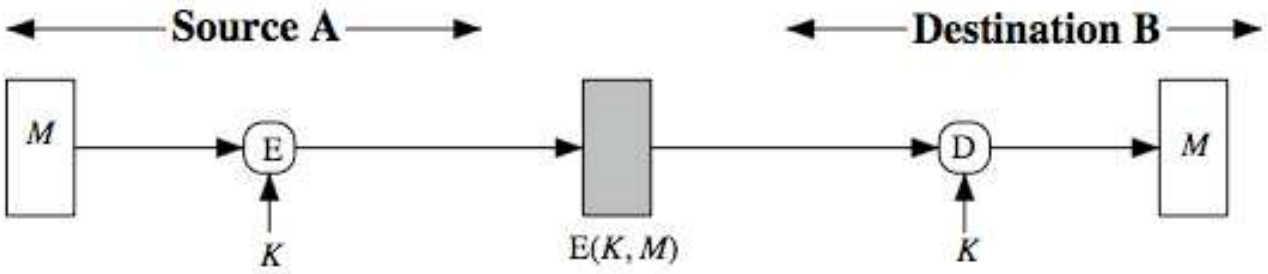
- disclosure
- traffic analysis
- masquerade
- content modification
- sequence modification
- timing modification
- source repudiation
- destination repudiation

Agenda

- Introduction
  - Message authentication requirements
  - Message authentication using encryption
  - MACs
  - HMAC authentication using a hash function
  - CMAC authentication using a block cipher
- Summary
- Test your understanding
- References

Symmetric Message Encryption

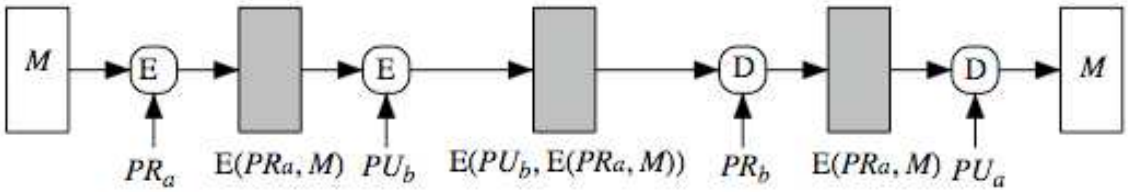
- encryption can also provides authentication
- if symmetric encryption is used then:
  - receiver know sender must have created it
  - since only sender and receiver know key used
  - know content cannot have been altered...
  - ... if message has suitable structure, redundancy or a suitable checksum to detect any changes



(a) Symmetric encryption: confidentiality and authentication

Public-Key Message Encryption

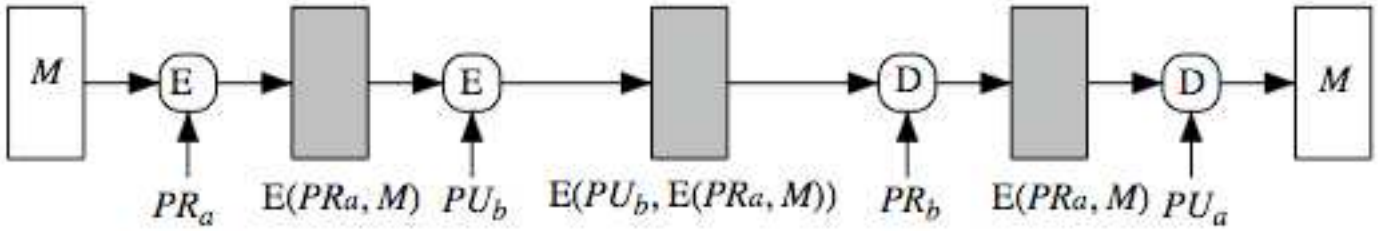
- if public-key encryption is used:
  - encryption provides no confidence of sender
    - since anyone potentially knows public-key
  - however if
    - sender **signs** message using their private-key
    - then encrypts with recipients public key
    - have both secrecy and authentication
  - again need to recognize corrupted messages
  - but at cost of two public-key uses on message



(d) Public-key encryption: confidentiality, authentication, and signature

Public-Key Message Encryption

- Dirty little detail on PKCS
  - Every time you encrypt, size expands
  - Due to protections in PKCS#1
- So signing (by encryption) then encrypting, the size is more than doubled!



(d) Public-key encryption: confidentiality, authentication, and signature

Agenda

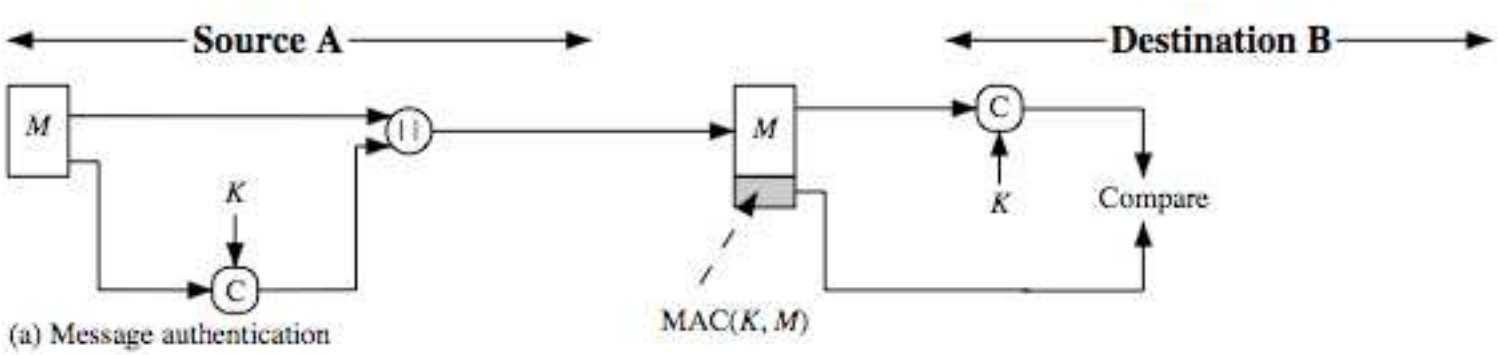
- Introduction
  - Message authentication requirements
  - Message authentication using encryption
  - MACs
  - HMAC authentication using a hash function
  - CMAC authentication using a block cipher
- Summary
- Test your understanding
- References

Message Authentication Code (MAC)

- generated by an algorithm that creates a small fixed-sized block
  - depending on both message and secret key
  - like encryption though need not be reversible
- appended to message as a "**signature"**
- receiver performs same computation on message and checks it matches the MAC
- provides assurance that message is unaltered and comes from sender

Message Authentication Code

- a small fixed-sized block of data
  - generated from message + secret key
  - MAC = C(K,M)
  - appended to message when sent



(a) Message authentication

Message Authentication Code

- An alternative authentication technique involves the use of a secret key to generate a small fixed- size block of data, known as a cryptographic checksum or MAC that is appended to the message.
- This technique assumes that two communicating parties, say A and B, share a common secret key *K*.
- When A has a message to send to B, it calculates the MAC as a function of the message and the key: *MAC = C(K, M).*
- The message plus MAC are transmitted to the intended recipient.
- The recipient performs the same calculation on the received message, using the same secret key, to generate a new MAC.

Message Authentication Code

- The received MAC is compared to the calculated MAC.
- If we assume that only the receiver and the sender know the identity of the secret key, and if the received MAC matches the calculated MAC,
  - then the receiver is assured that the message has not been altered, is from the alleged sender, and if the message includes a sequence number then the receiver can be assured of the proper sequence because an attacker cannot successfully alter the sequence number.
- A MAC function is similar to encryption.
- One difference is that the MAC algorithm need not be reversible, as it must for decryption.
- In general, the MAC function is a many-to-one function.

Message Authentication Codes

- as shown the MAC provides authentication
- can also use encryption for secrecy
  - generally use separate keys for each
  - can compute MAC either before or after encryption
  - is generally regarded as better done before, but see Generic Composition

Message Authentication Codes

- why use a MAC?
  - sometimes only authentication is needed
  - sometimes need authentication to persist longer than the encryption (e.g. archival use)
- note that a MAC is not a digital signature
  - Does NOT provide non-repudiation

MAC Properties

- a MAC is a cryptographic checksum

  MAC = $C_K(M)$

  - condenses a variable-length message M
  - using a secret key K
  - to a fixed-sized authenticator
- is a many-to-one function
  - potentially many messages have same MAC
  - but finding these needs to be very difficult

Requirements for MACs

- taking into account the types of attacks
- need the MAC to satisfy the following:
  1. knowing a message and MAC, is infeasible to find another message with same MAC
  2. MACs should be uniformly distributed
  3. MAC should depend equally on all bits of the message

Security of MACs

- like block ciphers have:
- **brute-force** attacks exploiting
  - strong collision resistance hash have cost $2^{m/2}$
    - 128-bit hash looks vulnerable, 160-bits better
  - MACs with known message-MAC pairs
    - can either attack keyspace (cf. key search) or MAC
    - at least 128-bit MAC is needed for security

Security of MACs

- **cryptanalytic attacks** exploit structure
  - like block ciphers want brute-force attacks to be the best alternative
- more variety of MACs so harder to generalize about cryptanalysis

Keyed Hash Functions as MACs

- want a MAC based on a hash function
  - because hash functions are generally faster
  - crypto hash function code is widely available
- hash includes a key along with message
- original proposal:

  KeyedHash = Hash(Key|Message)
  - some weaknesses were found with this
- eventually led to development of HMAC

Problem with Keyed Hash

- **KeyedHash = Hash(Key|Message)**
- Recall hash function works on blocks
- Let M = Key | Message | Padding and M

  $M=M_1\ M_2\ \ldots\ M_L$, where $|M_i|$ = Blocksize

  $Hash=H(H(\ldots H(H(IV,M_1),M_2),\ldots,M_L)$
- But can add extra block(s) M_{L+1} by

  $Hash'=H(Hash,M_{L+1})$
- Unless formatting prevents it…

  … but still best to use HMAC!

Agenda

- Introduction
- Message authentication requirements
- Message authentication using encryption
- MACs
- HMAC authentication using a hash function
- CMAC authentication using a block cipher
- Summary
- Test your understanding
- References

HMAC Design Objectives

- use, without modifications, hash functions
- allow for easy replacement of embedded hash function
- preserve original performance of hash function without significant degradation
- use and handle keys in a simple way.
- have well understood cryptographic analysis of authentication mechanism strength

HMAC

- specified as Internet standard RFC2104
- uses hash function on the message:

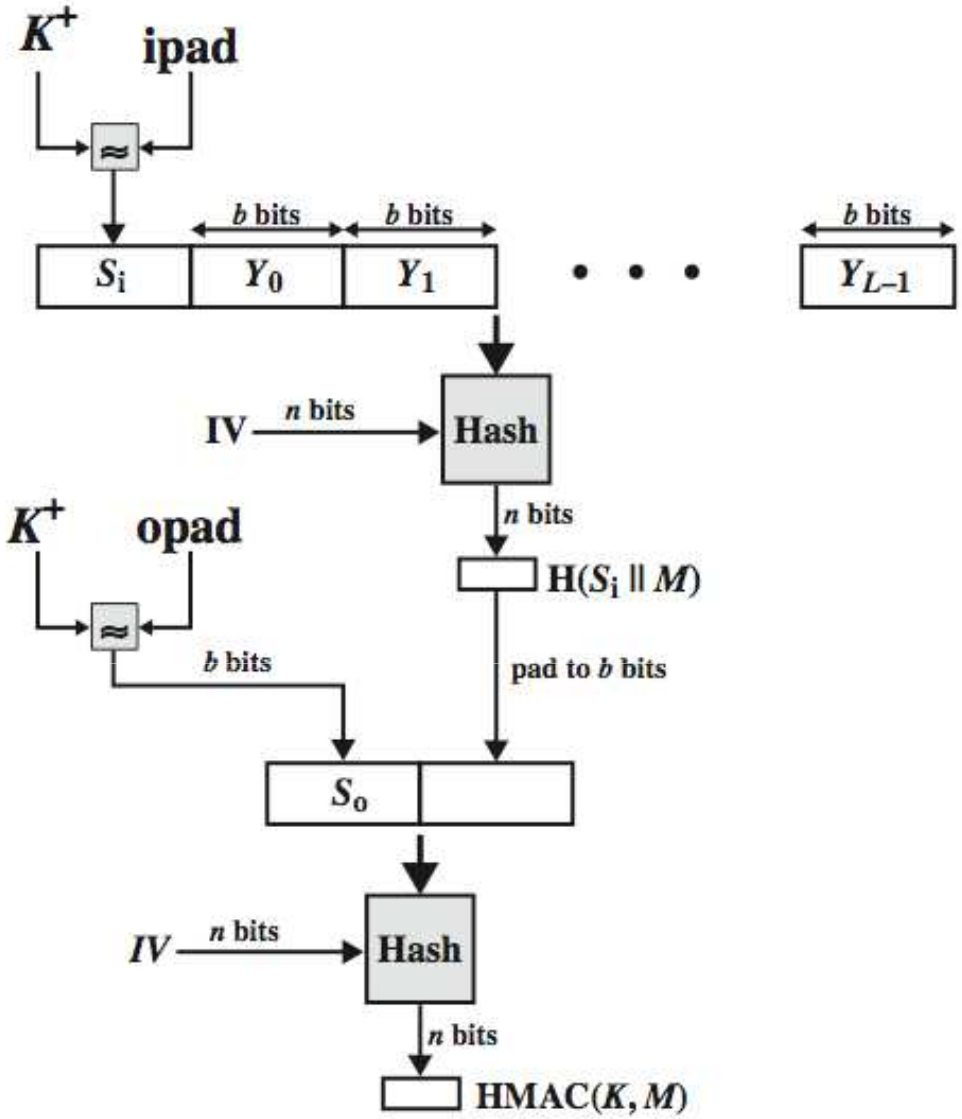
$$HMAC_K(M) = Hash[(K^+ \text{ XOR opad}) \,||\, Hash[(K^+ \text{ XOR ipad}) \,||\, M)]\,]$$

  - where K^+ is the key padded out to block size
  - **opad, ipad** are specified padding constants
- overhead is just 3 more hash block calculations than the message needs alone
- any hash function can be used
  - eg. MD5, SHA-1, RIPEMD-160, Whirlpool

HMAC

- The idea of a keyed hash evolved into HMAC, designed to overcome some problems with the original proposals.
- It involves hashing padded versions of the key concatenated with the message, and then with another outer hash of the result prepended by another padded variant of the key.
- The hash function need only be used on 3 more blocks than when hashing just the original message (for the two keys + inner hash).
- HMAC can use any desired hash function, and has been shown to have the same security as the underlying hash function.

HMAC Overview



HMAC

The overall operation of HMAC:

- $HMAC_K = Hash[(K^+ \text{ XOR } opad) \,||\, Hash[(K^+ \text{ XOR } ipad) \,||\, M)]$

where:

- K^+ is K padded with zeros on the left so that the result is b bits in length
- ipad is a pad value of 36 hex repeated to fill block
- opad is a pad value of 5C hex repeated to fill block
- M is the message input to HMAC (including the padding specified in the embedded hash function)
- Note that the XOR with ipad results in flipping one-half of the bits of *K.*
- Similarly, the XOR with opad results in flipping one-half of the bits of K, but a different set of bits.
- In effect, pseudorandomly generated two keys from K.
- HMAC should execute in approximately the same time as the embedded hash function for long messages.

HMAC Security

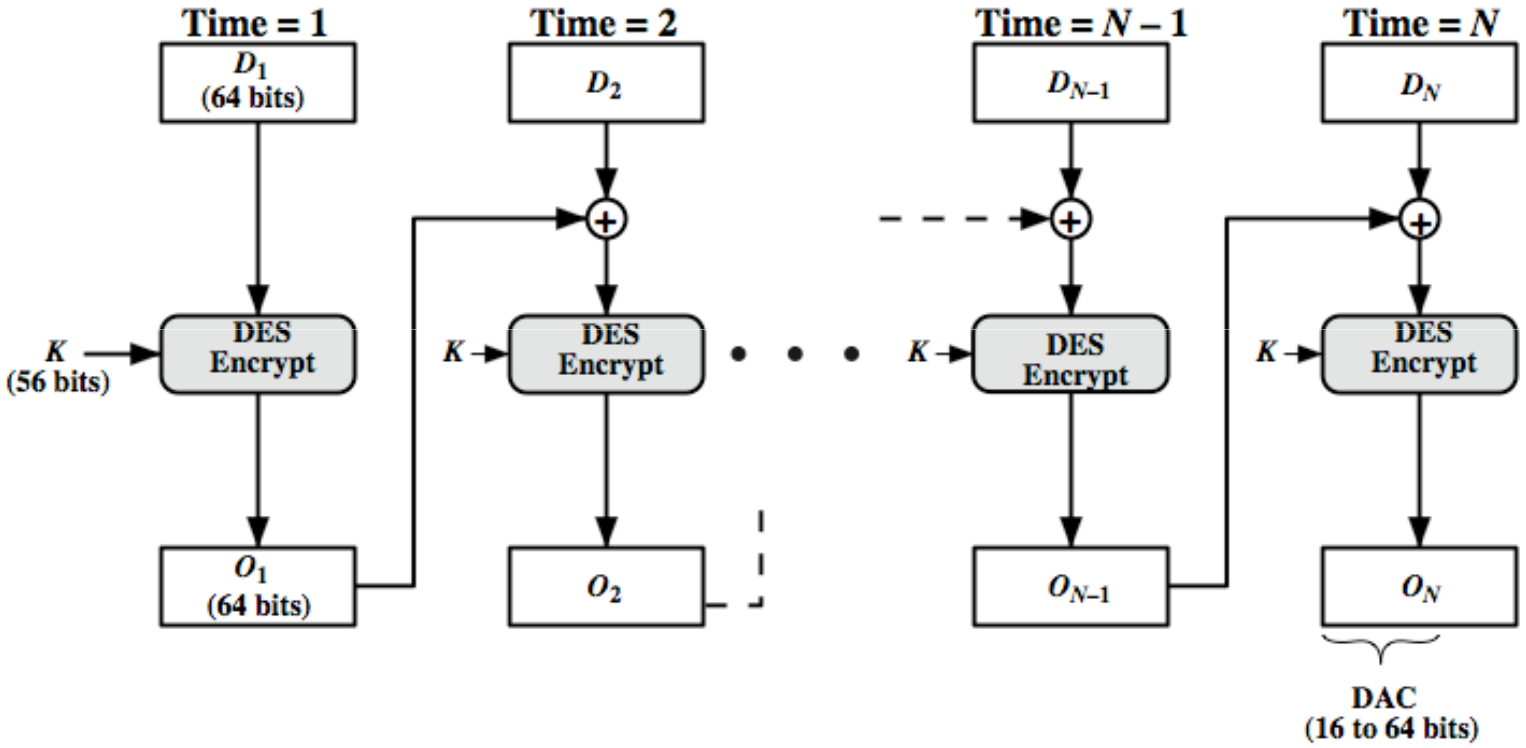
- proved security of HMAC relates to that of the underlying hash algorithm
- attacking HMAC requires either:
  - brute force attack on key used
  - birthday attack (but since keyed would need to observe a very large number of messages)
- choose hash function used based on speed verses security constraints

Using Symmetric Ciphers for MACs

- can use any block cipher chaining mode and use final block as a MAC
- **Data Authentication Algorithm (DAA)** is a widely used MAC based on DES-CBC
  - using IV=0 and zero-pad of final block
  - encrypt message using DES in CBC mode
  - and send just the final block as the MAC
    - or the leftmost M bits (16≤M≤64) of final block
- but final MAC is now too small for security…

… can use message blocks in reverse order…

Data Authentication Algorithm



Data Authentication Algorithm

- The Data Authentication Algorithm, based on DES, has been one of the most widely used MACs for a number of years.
- The algorithm is both a FIPS publication (FIPS PUB 113) and an ANSI standard (X9.17).
- However, security weaknesses in this algorithm have been discovered and it is being replaced by newer and stronger algorithms.
- The algorithm is shown here in Stallings Figure 12.7, and can be defined as using the cipher block chaining (CBC) mode of operation of DES, with an initialization vector of zero, and 0-pad of the final block if needed.
- Resulting MAC can be 16-64 bits of the final block.
- But, this is now too small for security.

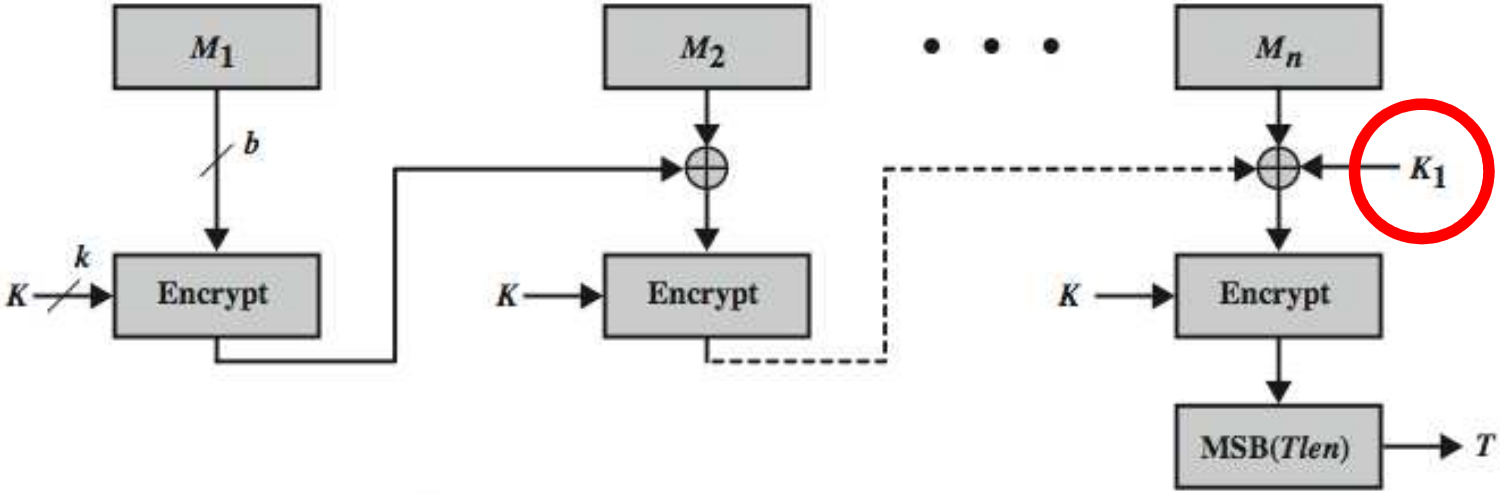
Agenda

- Introduction
  - Message authentication requirements
  - Message authentication using encryption
  - MACs
  - HMAC authentication using a hash function
  - CMAC authentication using a block cipher
- Summary
- Test your understanding
- References

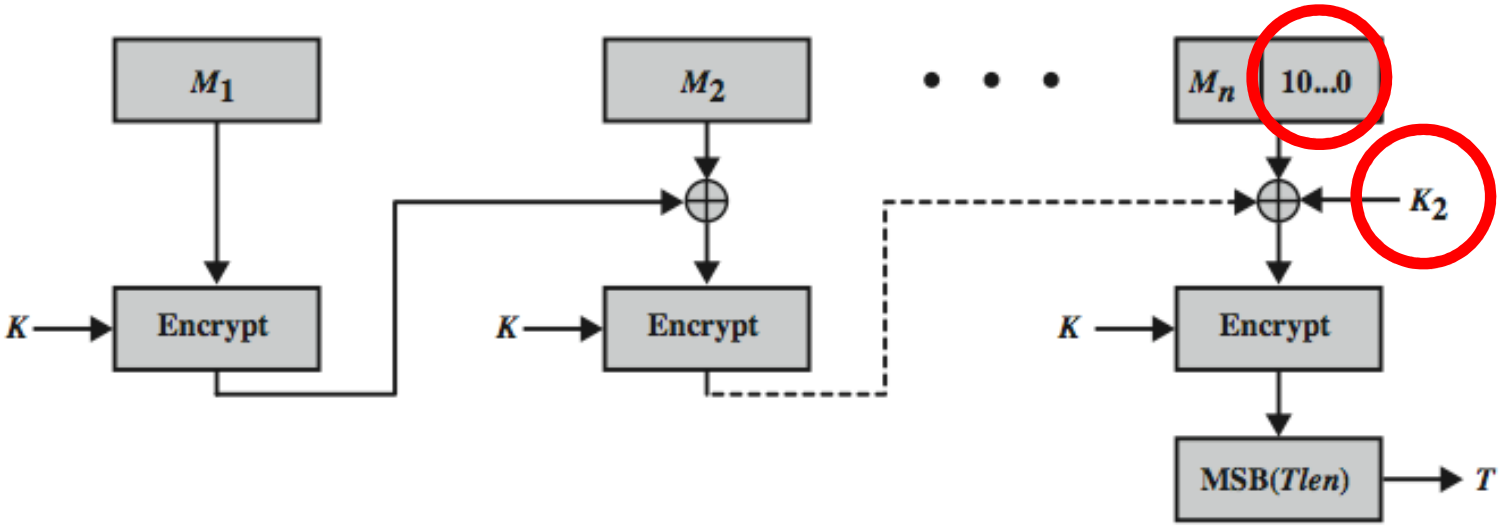
CMAC

- previously saw the DAA (CBC-MAC)
- widely used in govt & industry
- but has message size limitation
- can overcome using 2 keys & padding
- thus forming the Cipher-based Message Authentication Code (CMAC)
- adopted by NIST SP800-38B

CMAC Overview



(a) Message length is integer multiple of block size



(b) Message length is not integer multiple of block size

CMAC

- It uses the blocksize of the underlying cipher (ie 128-bits for AES or 64-bits for triple-DES).
- The message is divided into n blocks M1..Mn, padded if necessary.
- The algorithm makes use of a k-bit encryption key K and an n-bit constant K1 or K2 (depending on whether the message was padded or not).
- For AES, the key size k is 128,192, or 256 bits; for triple DES, the key size is 112 or 168 bits.
- The two constants K1 & K2 are derived from the original key K using encryption of 0 and multiplication in GF(2^n)

Agenda

- Introduction
  - Message authentication requirements
  - Message authentication using encryption
  - MACs
  - HMAC authentication using a hash function
  - CMAC authentication using a block cipher
- Summary
- Test your understanding
- References

Summary

- have considered:
  - message authentication requirements
  - message authentication using encryption
  - MACs
  - HMAC authentication using a hash function
  - CMAC authentication using a block cipher

Agenda

- Introduction
  - Message authentication requirements
  - Message authentication using encryption
  - MACs
  - HMAC authentication using a hash function
  - CMAC authentication using a block cipher
- Summary
- Test your understanding
- References

Test your understanding

1) What are the requirement of MAC?
2) Explain HMAC.
3) Explain CMAC.

Agenda

- Introduction
  - Message authentication requirements
  - Message authentication using encryption
  - MACs
  - HMAC authentication using a hash function
  - CMAC authentication using a block cipher
- Summary
- Test your understanding
- References

References

1. William Stallings, Cryptography and Network Security, 6th Edition, Pearson Education, March 2013.
2. Charlie Kaufman, Radia Perlman and Mike Speciner, “Network Security”, Prentice Hall of India, 2002.