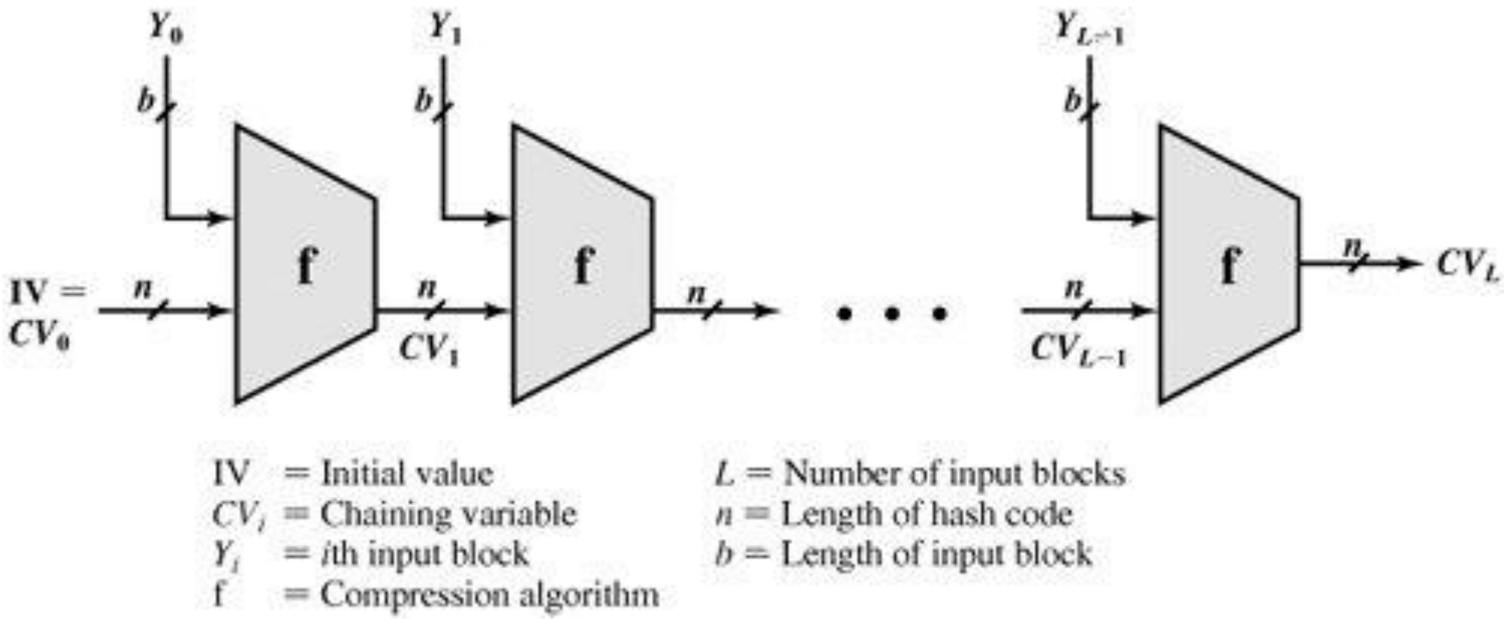
Fundamentals of Information & Network Security
ECE 471/571

Lecture #22: Hash Constructions and Integrity Check

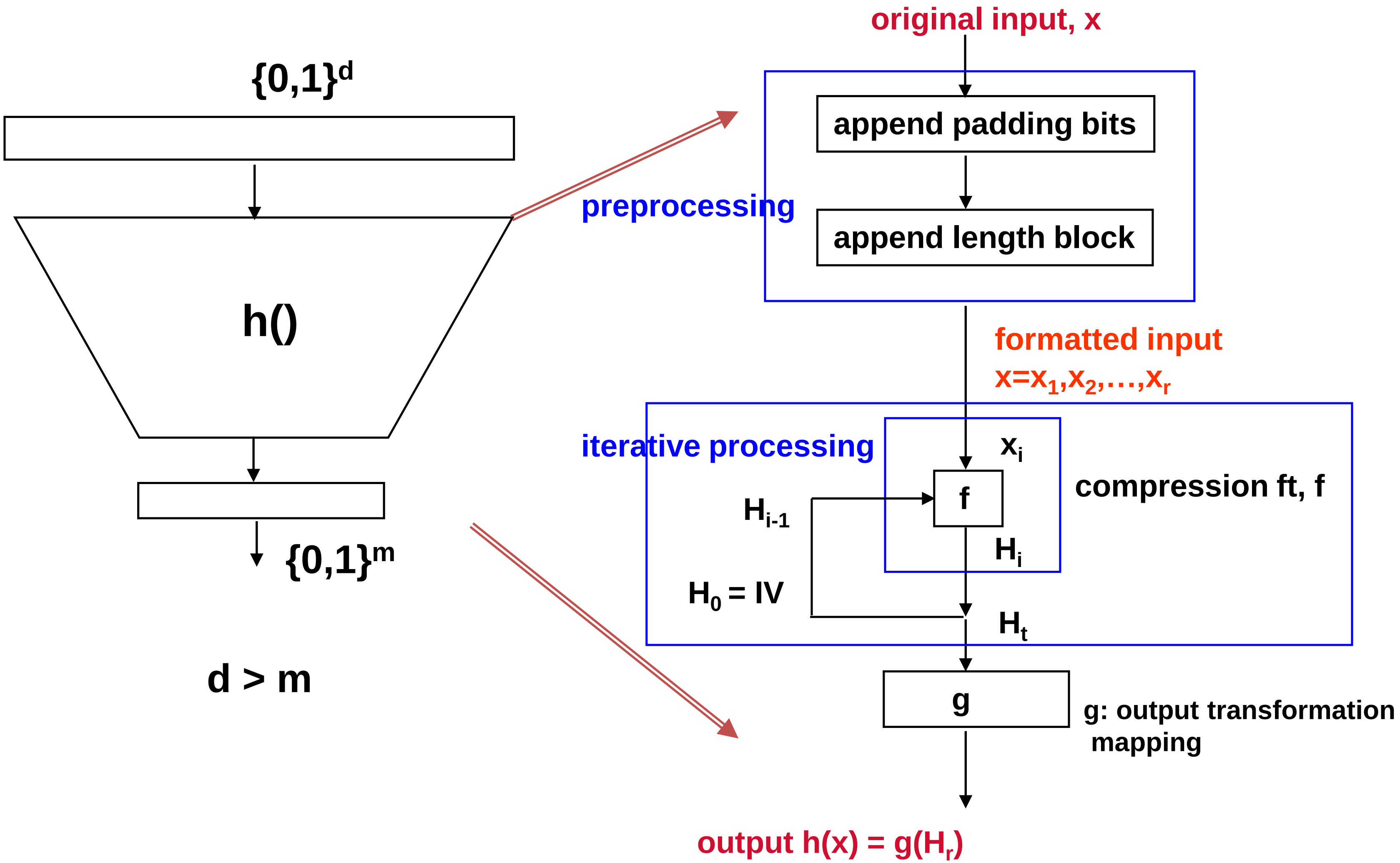
Instructor: Ming Li
Dept of Electrical and Computer Engineering
University of Arizona

Iterated Hash Functions



- Repeated use of a compression function, f, that takes two inputs (the chaining variable and input block), and produces an n-bit output

General Construction



MD2, MD4, MD5

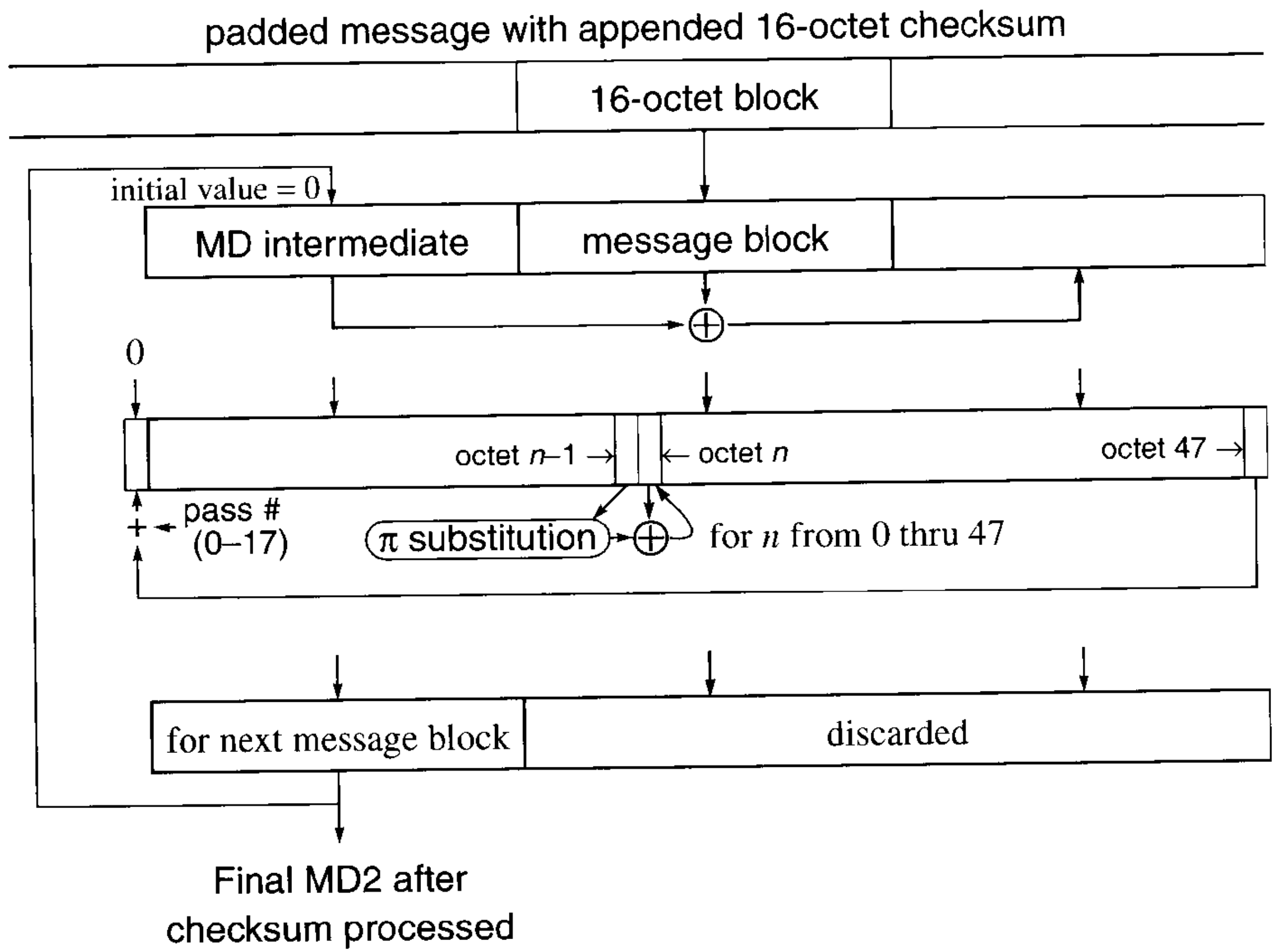


Figure 5-6. MD2 Final Pass

SHA-1

- Input: arbitrary number of bits ($\leq 2^{64}$)
- Output: 160 bits
- Step 1: Pad the message to be a multiple of 512 bits (16 words, 64 octets)
- Step 2: Process the message, 512 bits at a time, to produce the message digest

Comparison of MD5 and SHA-1

- SHA-1 160-bit, MD5 128-bit; SHA-1 is more secure against brute-force attacks
- MD5 is considered broken in year 2004
- SHA-1 involves more stages (80) and bigger buffer; SHA-1 executes more slowly than MD5
- SHA-1 is considered "broken" in year 2005
  - Collision has been discovered in the full version in 2^69 hash operations
  - No longer used after 2010

Performance

AMD Opteron 8354 2.2 GHz processor running 64-bit Linux

Algorithm	Length	Security	Speed (MB/s)
MD4	128	< 28	
MD5	128	< 64	335
SHA-1	160	< 80	192
SHA-2	256	128	139
	384	192	154
	512	256	
SHA-3	256	128	
	384	192	
	512	256	

https://www.cryptopp.com/benchmarks-amd64.html
https://en.wikipedia.org/wiki/Secure_Hash_Algorithm

Message Authentication Code (MAC)

- MD(m) ?
- MD(K_{AB}|m): only the one who knows the secret can compute/verify
- Problem?

The Problem with keyed hash *h(Key|m)*

- A feature of message digest algorithms
  - In order to compute the message digest through chunk *n*, all that you need to know is the message digest through chunk *n-1*, plus the chuck *n* of the padded message.
- An Attack
  - Someone gets m, and digest(Key|m)
  - He first pads m according the used hash function, and then adds another message M at the end. The result is m|pad|M.
  - digest(Key|m|pad |M) can be calculated from digest(Key|m), which is the intermediate digest.

Solutions

- Use h(m|Key)
- HMAC

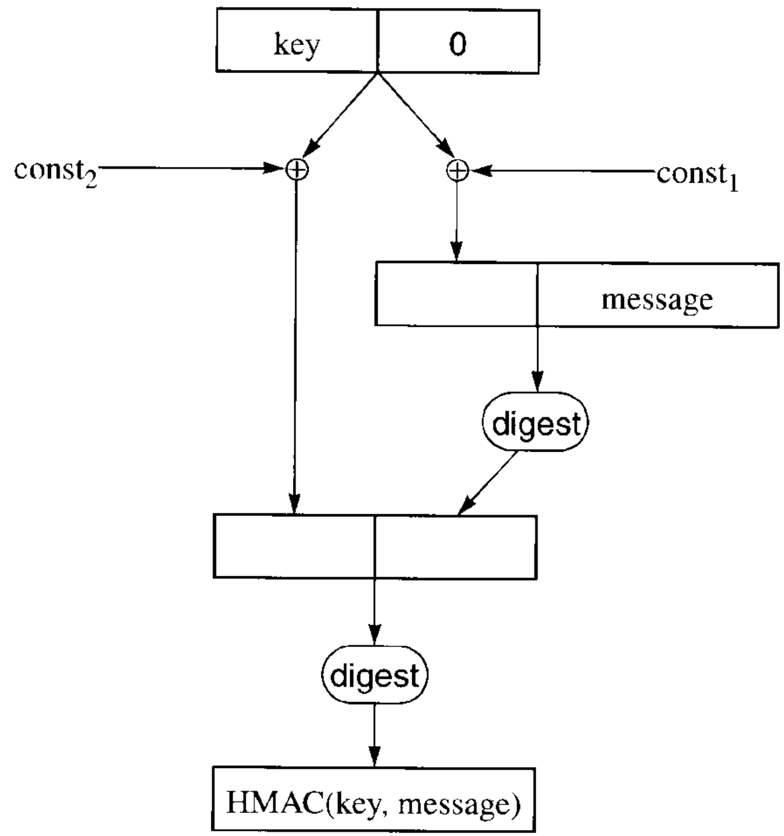
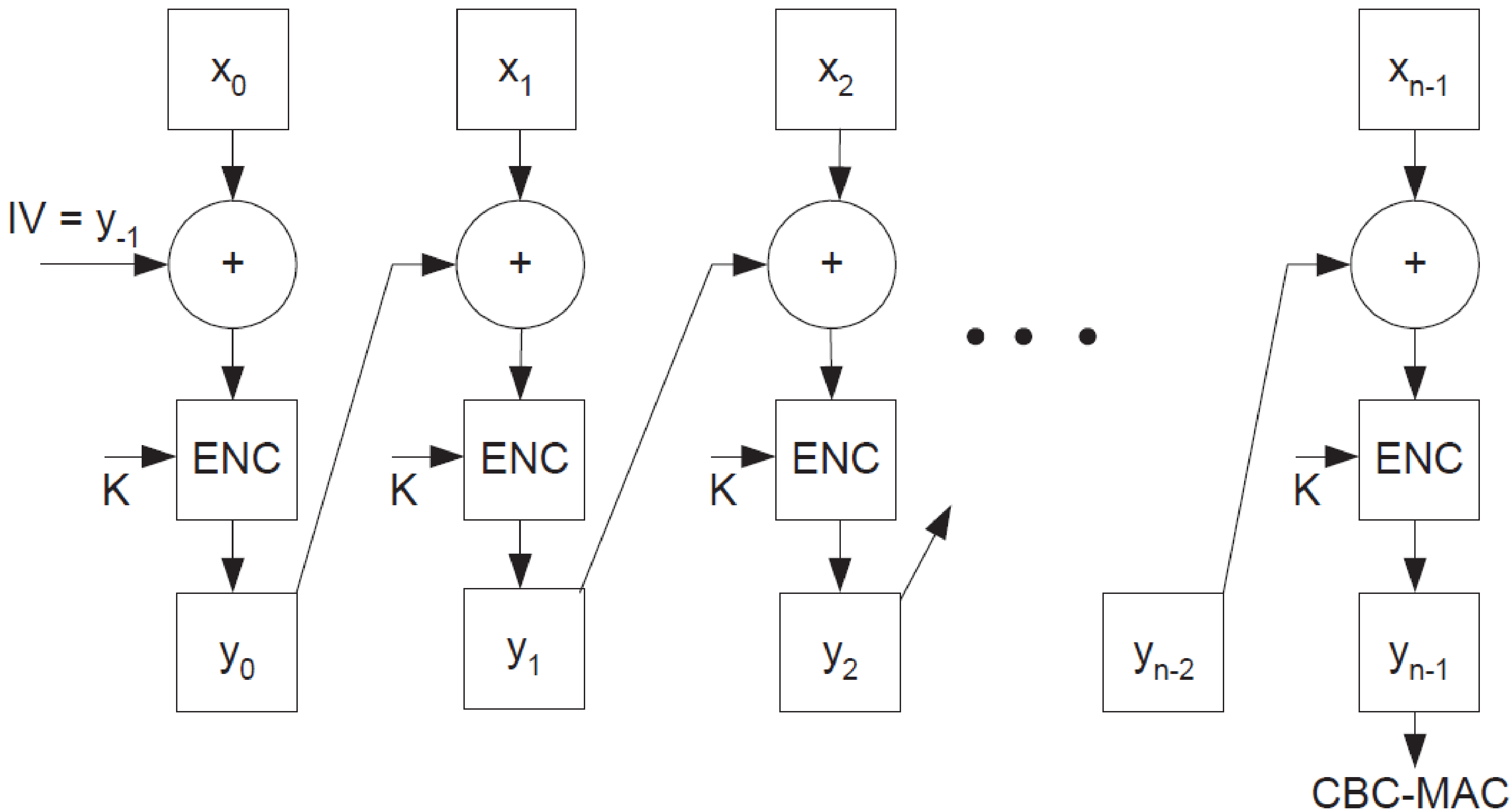


Figure 5-10. HMAC

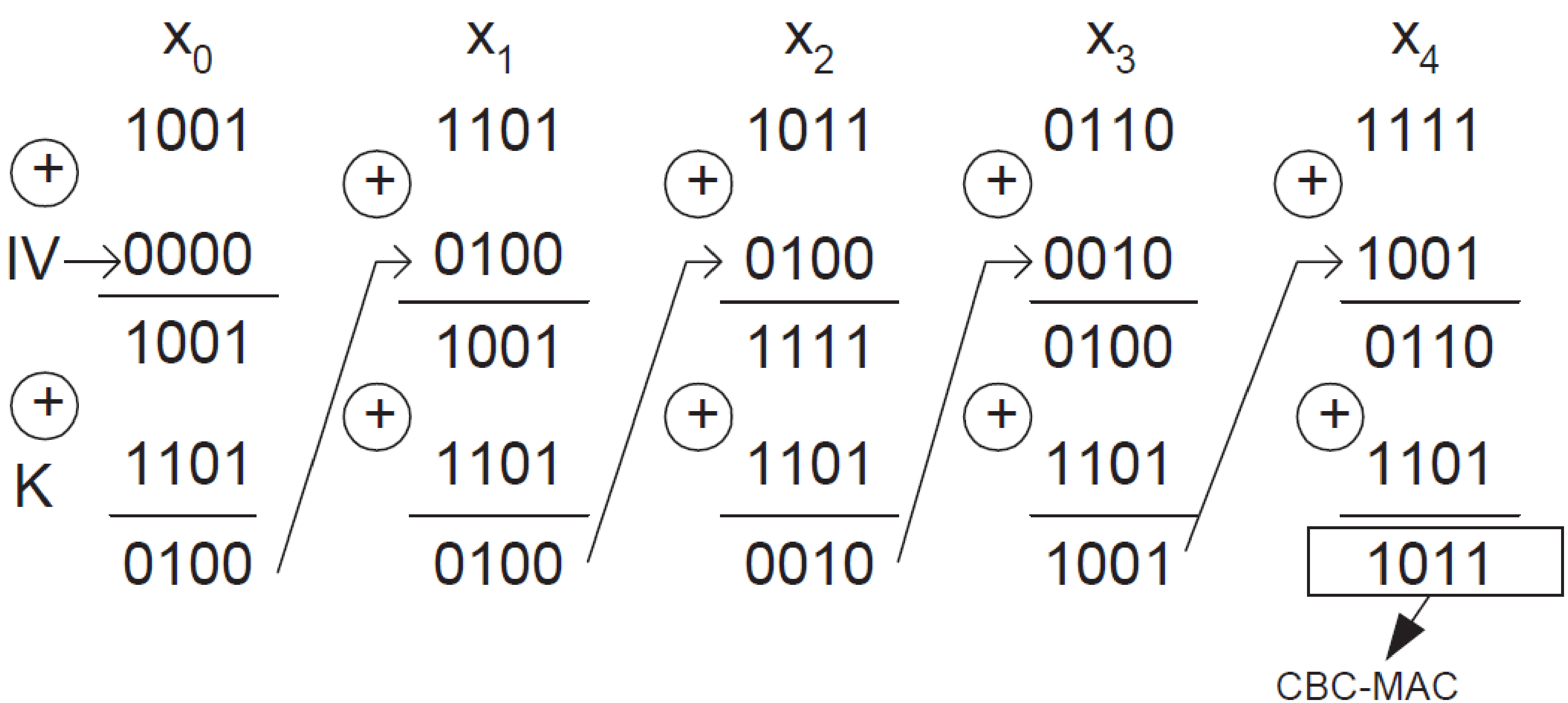
$$\mathrm{HMAC}_K(x) = \mathrm{SHA\text{-}1}((K \oplus opad) \mid\mid \mathrm{SHA\text{-}1}((K \oplus ipad) \mid\mid x)).$$

CBC-MAC



Example

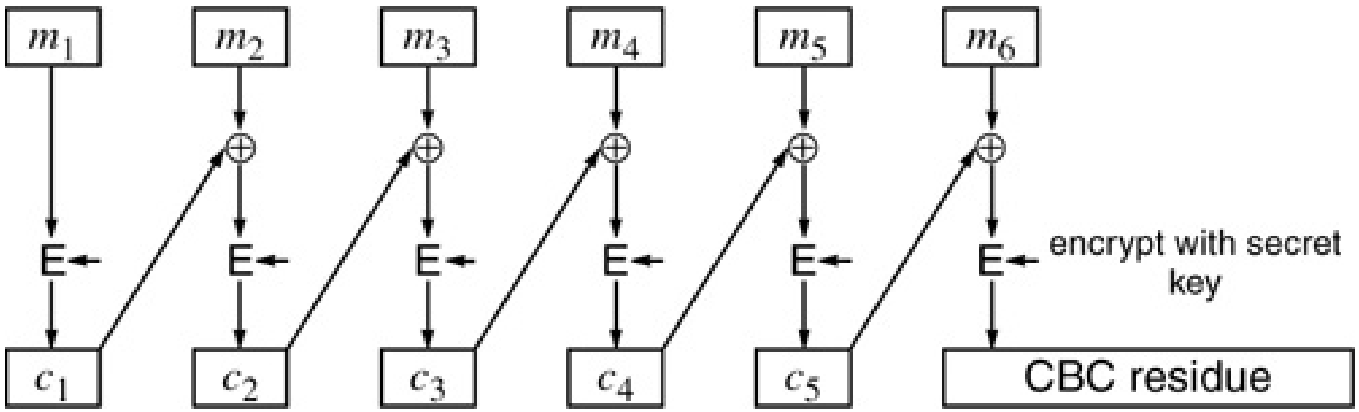
- How does Bob know the message length?
- Is this a good CBC-MAC?



Alice sends to Bob: 10011101101101101111**1011**

Integrity: Generating MACs

- Protect against undetected modifications
- Plaintext + CBC residue (when message not secret)



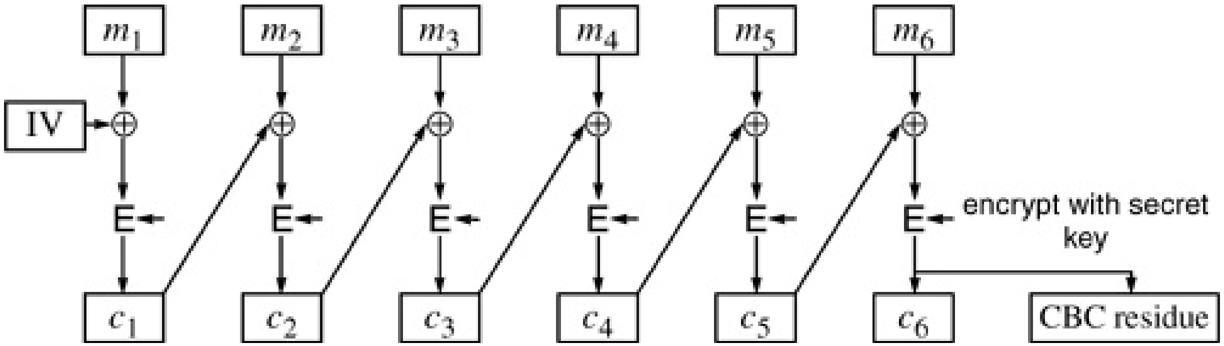
Privacy and Integrity Together (1)

- Privacy: CBC encryption
  Integrity: CBC residue

- Ciphertext + CBC residue ?
- Encrypt {plaintext + CBC residue} ?
- Encrypt {plaintext + CRC} ?

Ciphertext + CBC Residue

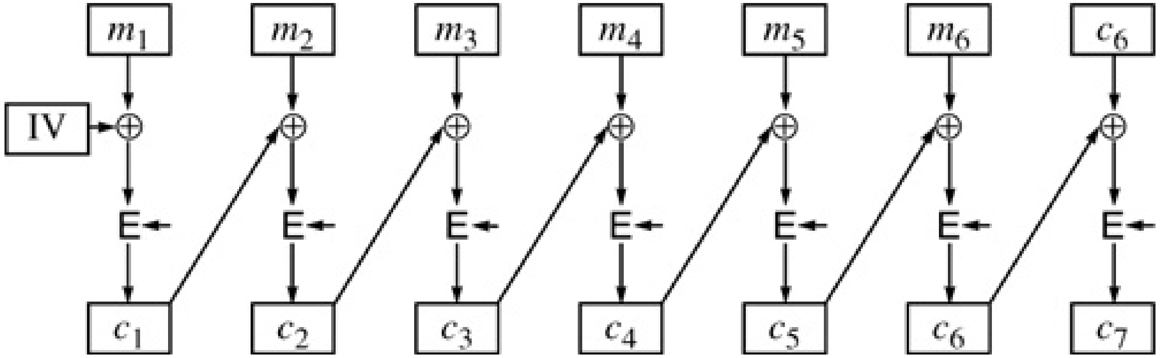
Figure 4-12. Cipher Block Chaining Encryption plus CBC Residue



- Problem?

Encrypt {plaintext + CBC residue}

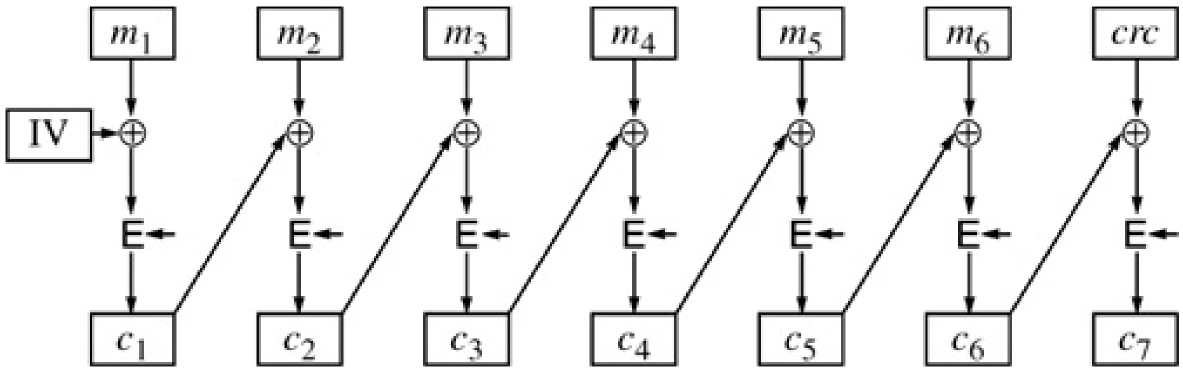
Figure 4-13. Cipher Block Chaining Encryption of Message with CBC Residue



- Problem?

Encrypt {plaintext + CRC}

Figure 4-14. Cipher Block Chaining Encryption of Message with CRC



- Longer CRC maybe Okay

Privacy and Integrity: The Do's

- Privacy: CBC encryption + Integrity: CBC residue, but with different keys
- CBC + weak cryptographic checksum
- CBC + CBC residue with related keys
- CBC + cryptographic hash: keyed hash preferred
- ……