Computer Security

10r. Recitation – assignment & concept review

Paul Krzyzanowski

Rutgers University

Spring 2018

1. What is a necessary condition for perfect secrecy?

Claude Shannon proved that a cipher has perfect secrecy *if and only if* there are as many possible keys as possible plaintexts, and every key is equally likely; so the one-time pad is the only kind of system which offers perfect secrecy"

See page 133 of the text

2. What is a running key?

- A key that repeats to make it as long as the message.

3. What are the three properties of hash functions?

1. One-wayness. Given knowledge of an input *x* we can easily compute the hash value *h(x)*, but it is very difficult given the hash value *h(x)* to find a corresponding preimage *x* if one is not already known.
2. The output will not give any information at all about even part of the input.
3. It is hard to find collisions, that is, different messages *M1* ≠ *M2* with *h(M1)* = *h(M2)*.

See page 141 of the text

3. Discussion: properties of cryptographic hash functions

There are actually more than three properties.
Properties include:

- They produce **fixed-length output** for arbitrary length inputs
- They are **deterministic**: you always get the same hash for the same message
- They are **one-way functions (also called *pre-image resistance*, or *hiding*)**
  - Given *H*, it should be difficult to find *M* such that *H=hash(M)*,
- Their **output should not give any information about any of the input**
  - Like cryptographic algorithms, relies on *diffusion*
- They are **collision resistant**
  - Infeasible to find any two different strings that hash to the same value:
    Find *M, M'* such that *hash(M) = hash(M')*
- They are **efficient**
  - Computing a hash function should be computationally efficient
  - We'd like to be able to use them for every message we send, for example, without incurring a significant performance penalty.

4. How many messages would you have to try, on average, to find two that have a collision (hash to the same hash value) for an n-bit hash function?

- This is the birthday paradox (theorem)
- Answer is $2^{(n/2)}$

See page 142 of the text

4. Discussion: hash collisions

wikipedia

- Hash collisions can occur
- Pigeonhole principle
  - If you have 10 pigeons & 9 compartments, at least one compartment will have more than one pigeon
  - A hash is a fixed-size small number of bits (e.g., 256 bits = 32 bytes)
  - Every possible permutation of an arbitrary number of bytes cannot fit into every permutation of 32 bytes!

- Why are we not worried?
  - Because a 256-bit hash means that our data hashes to one of 2^{256} possible values.
  - The chances that something else hashes to that same value is 1 in 2^{256}, which is 1 in 1.58×10^{77}
    - The odds of winning the powerball are 1 in 2.9×10^{8} … 4×10^{68} times greater!

4. Discussion: hash collisions

- The Birthday Paradox
  - What are the odds that two students in a class of 30 will have the same birthday?
  - Answer: 70%
  - There's a 50% chance that two students in a class of 23 will have the same birthday
- Why is this counterintuitive?
  - We think of the problem as "what are the odds that there's a student in the class that has the same birthday *as one selected student*" not "*any two students*"
- Hash collisions
  - Same applies to hashes. If we have a large amount of files or messages, along with hashes of that data, the likelihood that two messages will have the same hash is about 1 in $2^{(n/2)}$ for an n-bit hash
  - So with a 256-bit hash function means there's a 1 in 2^{128} chance of finding a collision … but we're still comfortable with 1:3.4x10^{38} odds!

5. What is an s-box in a block cipher?

- S-box = substitution box
- Logically, a lookup table that permutes a set of input bits to a set of output bits in an invertible manner.

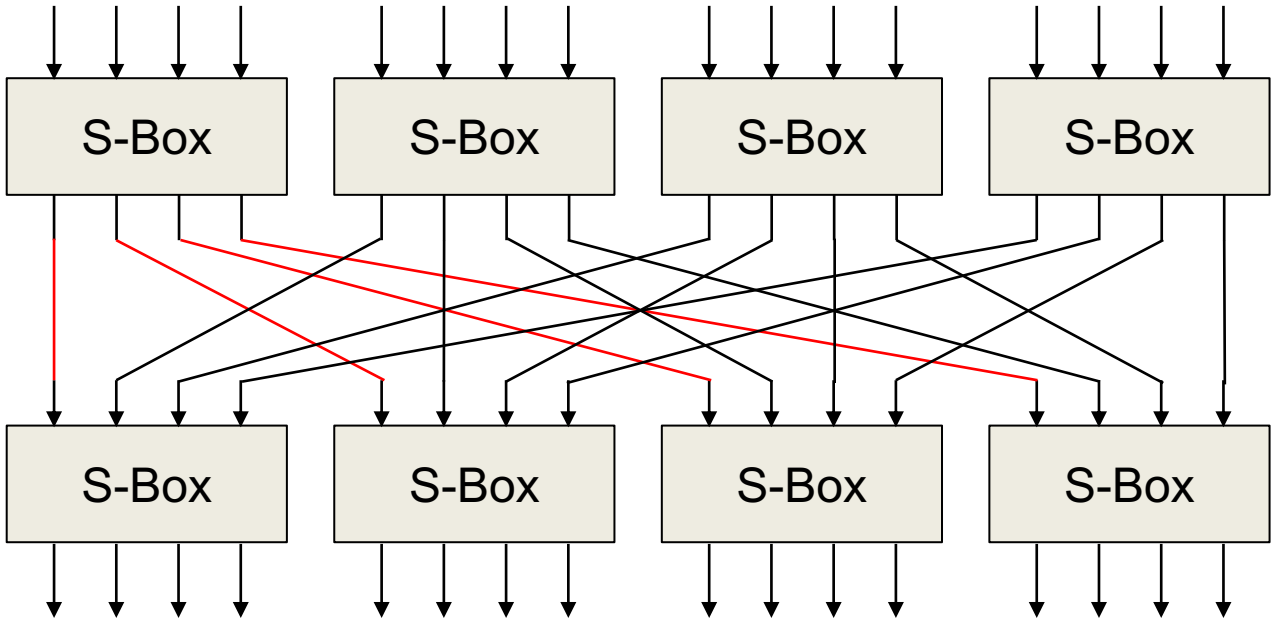
See page 149 of the text

5. Discussion: s-boxes

- Block ciphers
  - Encrypt a chunk of data at a time (rather than a byte at a time)
  - Versus stream ciphers
  - Essentially all symmetric block ciphers use SP Networks
- General goal: *Confusion and Diffusion*
  - Confusion = add secret key values
  - Diffusion = spread plaintext data throughout ciphertext block
- SP Networks: substitution and permutation
  - S-box = lookup table that maps a set of bits onto another set
  - Some bits of the key may select which s-box to use
  - … or some bits of the key might be used as input to the s-box

5. Discussion: s-boxes

- Encryption involves multiple rounds
  - The output of one set of s-box operations is used as input to the next round
- A simple 16-bit, 2-round SP-network from the text (p. 151):



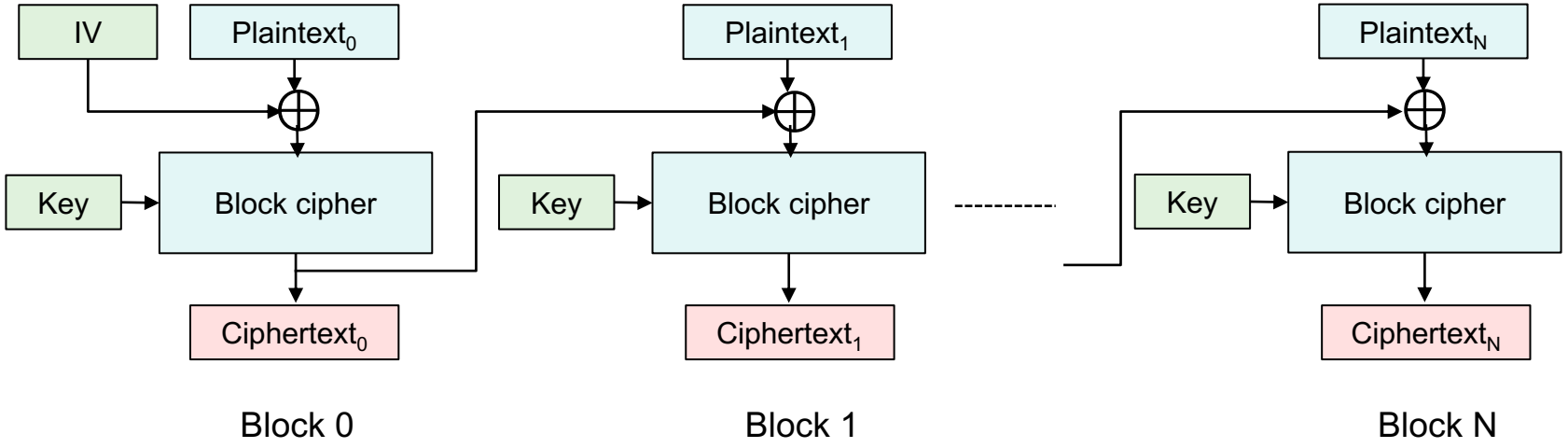
6. What is an advantage of counter encryption (CTR) over cipher block chaining (CBC)?

- Blocks can be encrypted in parallel
- With Cipher Block Chaining, encryption cannot be parallelized

See page 162 of the text

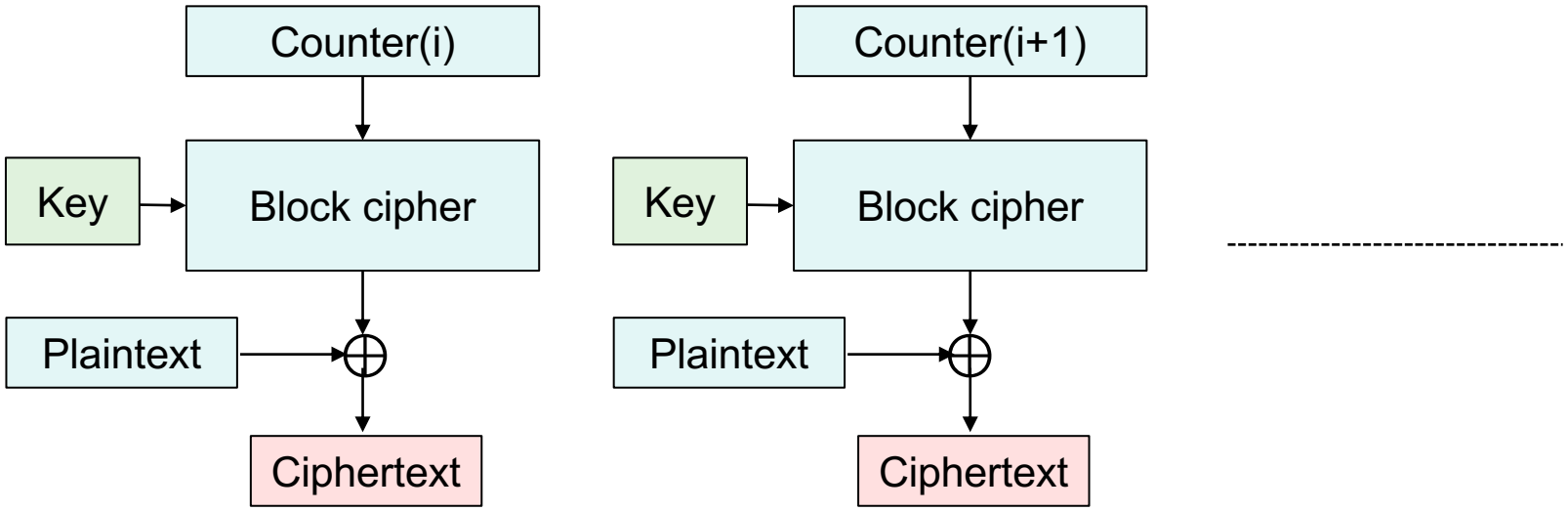
6. Discussion: CBC vs CTR modes

- Cipher Block Chaining (CBC)
  - Encryption of multiple blocks cannot be parallelized
  - The entire previous block of the cipher must be encrypted before the next block can be generated.
  - Starts with a random initialization vector (IV) = bunch of *k* random bits
    - Exclusive-or with first plaintext block – then encrypt the block
    - Take exclusive-or of the result with the next plaintext block



6. Discussion: CBC vs CTR modes

- Counter encryption (CTR)
  - Counter encryption requires encrypting a counter that is then added (XORed) to the plaintext.
  - It also ensures that there is no chance of seeing repeats: the use of identical outputs from previous results
  - Counter start is a non-secret random number that everyone knows, like the initialization vector (IV)



Conversation Isolation: Transport Layer SSL/TLS

We can't count on the security of the Internet

- Core IP protocols were not designed with security in mind
- Traffic can be redirected
  - For interception for modification or logging
  - For deception: adversary masquerades as the server
- What can we do … without changing the way IP works?
  - **Use virtual private networks – VPNs**
    - Provide an authenticated, and optionally encrypted, message stream between two networks
      - Treat entire IP packets as data
      - This data is sent via IP and has an authentication header (MAC) to ensure that it has not been modified … and is optionally encrypted
      - Transport mode: form of VPN that communicates between one host and a network
    - VPNs were designed to operate at the *network layer*
      - Connect networks together
  - Or provide this type of security at the ***transport layer***

Transport Layer Security

- Goal: provide a *transport layer* security protocol
- After setup, applications feel like they are using TCP sockets

SSL: Secure Socket Layer

- Created with HTTP in mind
  - Web sessions should be secure
  - Mutual authentication is usually not needed
    - Client needs to identify the server but the server won't know all clients
    - Rely on password authentication after the secure channel is set up
- SSL evolved to TLS (Transport Layer Security)
  - SSL 3.0 was the last version of SSL … and is considered insecure
  - We use TLS now … but often still call it SSL

TLS Protocol

- Goal
  - Provide authentication (usually one-way), privacy, & data integrity between two applications
- Principles
  - **Data encryption**
    - Use symmetric cryptography to encrypt data
    - Keys generated uniquely at the start of each session
  - **Data integrity**
    - Include a MAC with transmitted data to ensure message integrity
  - **Authentication**
    - Use public key cryptography & X.509 certificates for authentication
    - Optional – can authenticate 0, 1, or both parties
  - **Interoperability & evolution**
    - Support many different key exchange, encryption, integrity, & authentication protocols – negotiate what to use at the start of a session

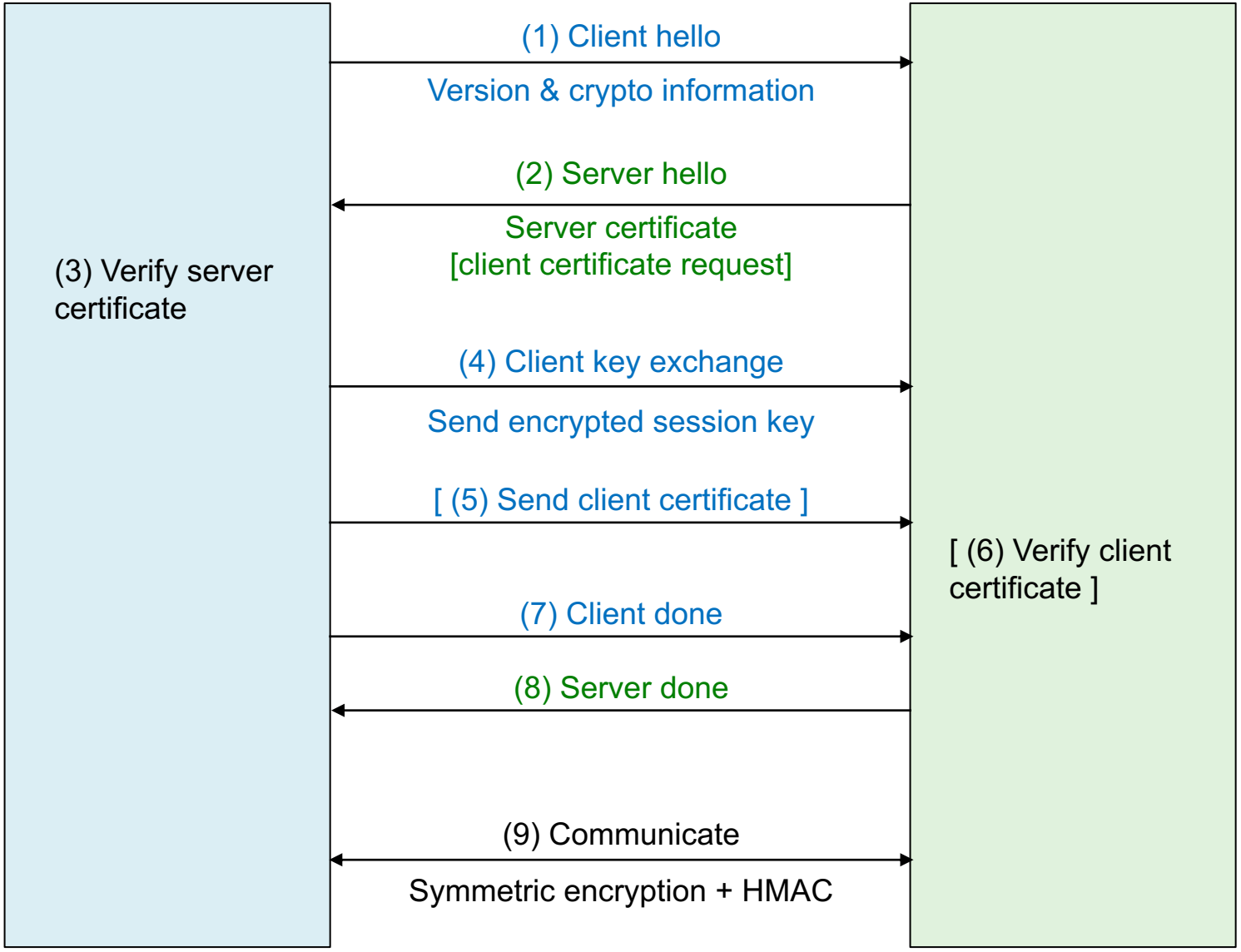
TLS Protocol & Ciphers

Two sub-protocols

1. Authenticate & establish key
2. Communicate
   - HMAC used for message authentication

- Key exchange
  - Public keys (RSA or Elliptic Curve)
  - Diffie Hellman keys
  - Ephemeral Diffie-Hellman keys (generated for each session)
  - Pre-shared key
- Data encryption
  - AES GCM, AES CBC, ARIA (GCM/CBC), ChaCha20-Poly1305, …
- Data integrity
  - HMAC-MD5, HMAC-SHA1, HMAC-SHA256/384, …

TLS Protocol



Benefits of TLS

- Benefits
  - Protects integrity of communications
  - Protects the privacy of communications
  - Validates the authenticity of the server (if you trust the CA)

Attacks on TLS

- Man-in-the-middle: BEAST attack in TLS 1.0
  - Attacker was able to see Initialization Vector (IV) for CBC and deduce plaintext (because of known HTML headers & cookies)
  - Fixed by using explicit IVs for each new block
- Man-in-the-middle: crypto renegotiation
  - Attacker can renegotiate the handshake protocol during the session to disable encryption
  - Proposed fix: have client & server verify info about previous handshakes
- THC-SSL-DoS attack
  - Attacker initiates a TLS handshake & requests a renegotiation of the encryption key – repeat over & over, using up server resources

Other problems with TLS

- Client authentication Problem
  - Client authentication is almost never used
    - Generating keys & obtaining certificates is not an easy process for users
    - Any site can request the certificate
      - User will be unaware their anonymity is lost
    - Moving private keys around can be difficult
      - What about public computers?
  - We usually rely on other authentication mechanisms
    - Usually user name and password
    - But no danger of eavesdropping since the session is encrypted
    - May use one-time passwords or two-factor authentication if worried about eavesdroppers at physical premises

The end