Cristina Nita-Rotaru

CS6740: Network security

Introduction: Class overview. Fundamentals.

1: Class overview

Turkish banks & government sites under ‘intense’ DDoS attacks on Christmas holidays

on December 25, 2015 |

DDoS | DDoS Attack Specialist | DDoS Defense | Defend Against DDoS | DoS Attacks

Turkey is suffering from a wave of cyber-attacks on financial and government websites which intensified over Christmas, resulting in the temporary disruption of credit card transactions.
A video released this week and attributed to Anonymous vowed retribution for Ankara’s alleged ties with ISIS.

The attacks on Turkish servers have been persistent in recent weeks, but on Christmas day Turkish banks suffered a website outage and reportedly saw sporadic disruption to credit card transactions. Isbank, Garanti and Ziraat Bank were among the targets, local media reported.

IoT Is A New Backdoor For DDoS Attacks

cyber attack cybersecurity cyberwar information security

One of the more unorthodox and interesting ways that hackers can perform a DDoS attack is through the backdoor of millions and millions of improbable devices from Fridges to Coffeepots. Using the fast emerging Internet of Things (IoT) hackers can take control of vast amount of hardware to send out their malicious attacks.

On many newly manufactured products: from smartphone controlled thermostats, smart aquariums, and GPS trackers on pets to other autonomous technology such as wind turbines and forest fire detection sensors, there is a built in provision for connection to the Internet. So much so, that HTML protocols for all sorts of weird and wonderful things exist; even coffeepots!

BGP routing incidents in 2014, malicious or not?

Posted by Andree Toonk - February 17, 2015 - BGPmon.net, Hijack - No Comments

Over the last year we have seen and written about numerous BGP routing incidents that looked out of the ordinary, straight-up suspicious or were just configuration mistakes. In this blog post we will highlight a few of them and look at the impact and cause of each of the observed incidents and try to determine if there was any malicious intent.

I presented the same data last week at NANOG 63, in San Antonio, a recording of this presentation can be found below:

BGP hijacking for monetary gain.

Bitcoin Mining Pools Targeted in Wave of DDOS Attacks

Stan Higgins | Published on March 12, 2015 at 18:54 GMT

NEWS

84 23 106 3

AntPool, BW.com, NiceHash, CKPool and GHash.io are among a number of bitcoin mining pools and operations that have been hit by distributed denial-of-service (DDOS) attacks in recent days.

The incidents appear to have begun in the first week of March. For example, on 11th March, AntPool owner Bitmain sent an email to customers disclosing the DDOS attacks and advising external pool users to set up failsafe pools in the event of an outage.

According to many of the companies affected by the incidents, those behind the attacks demanded payment in bitcoin in return for stopping the attacks.

What is network security

``Network Security is the process of taking physical and software preventative measures to protect the underlying networking infrastructure from unauthorized access, misuse, malfunction, modification, destruction, or improper disclosure, thereby creating a secure platform for computers, users and programs to perform their permitted critical functions within a secure environment.''

SANS Institute

Why study network security?

- JOB SECURITY

- Security is a major component of computer science with great impact on critical infrastructure and everyday life
- Network-enabled devices and gadgets presence in everyday life increased and will continue to increase

What is this course about?

- Learn to think about (network) security:
  - Threats, defenses, policies
  - Software, human and environment factors
- Think as an attacker:
  - Learn to identify threats
- Think as a security designer:
  - Learn how to prevent attacks and/or limit their consequences
  - Understand and apply security principles
  - Learn tools that can defend against specific attacks, no silver-bullet solution

Prerequisites

- Strong systems and networking background
  - Assembly language and memory layouts
  - The ISO/OSI network stack, BGP, DNS, and HTTP
- Fluency in many languages
  - C/C++
  - HTML and Javascript
  - Python or some other scripting language
- Linux command line proficiency
- Computer security and cryptography fundamentals

Course outline

- Security and privacy goals for network protocols. Fundamentals.
- Internet:
  - Link layer and transport security: ARP, TCP, IPSEC, TLS, HTTPS, QUIC
  - Naming: DNS, DNSSEC, RPKI
  - Routing: OSPF, BGP
  - Web security
- Wireless networks:
  - 802.11
  - Cellular
  - Other: bluetooth, WIMAX, IoT
- Anonymity: mixnets, onion routing, TOR, location privacy
- Emergent networks: vehicular, car, SDN

Course information

- Meetings
  - Tu 6:00-9:00 221 Hayden Hall
- Professor contact info:
  - Office: 258 WVH
  - Email: c.nitarotaru
  - Office hours: 4pm – 6pm before class and by appointment
- Class webpage

  http://cnitarot.github.io/courses/ds_Fall_2016/index.html
- Use Piazza for questions and postings
- Hw and projects posted on piazza

Grading policy

- Written assignments 10%
- Programming projects 35%
- Midterm 20%
- Final 30%
- Class participation 5%

- There is no curve for grades

Written assignments

- **Purpose of the written assignments is to prepare you for the midterm and final exams**
  - **Read the material before solving them and solve them with closed books and notebooks**
- 3 written theoretical assignments
- Homework is individual
- Homework must be typed – PDF submission format only
- For submission, follow the information in the homework description

Programming projects

- **Purpose of the programming projects is to help you understand practical aspects of things discussed in class**
  - **Read all material in class and the description of the project in details before starting**
  - **Make sure you understand the observed results for the items you are asked to investigate for the reports**
- 3 programming projects
- Programming projects are individual
- All the code must be from scratch
- Use the VMs/machines specified in the project description

Late policy

- Each of you gets 5 LATE DAYS that can be used any way you want for homework and projects; you do not need to let us know if you plan to take any late day; just submit late
  - Keep track of your late days used
  - 20% off from grade obtained per day late
- Do not wait till the last moment
- Follow the requirements from project description to see how to submit
- **Assignments are due at 9:59:59 pm, no exceptions**
  - **1 second late = 1 hour late = 1 day late**

Midterm and final exams

- Midterm is two hours
  - Preliminary date is Feb 23 in class
- Final is two hours
- We will have review for the midterm and final
- We will discuss the midterm solutions in class
- All exams are closed books, closed notebooks
- No electronic devices, laptops, tablets, phones, etc
- Exams cover everything, including written assignments and projects

Class attendance and notes

- Your are strongly recommended to attend and take notes
- If you miss class is your responsibility to go through the covered material on your own
- Slides will be made available online before lecture;
- There is no book for the class, there will be assigned reading from papers and other online materials
- Class participation is 5% of your grade
  - Be active on Piazza
  - Ask questions in class
  - Answer questions in class

Regrading

- YOU HAVE 1 WEEK to ASK for REGRADING of a homework, project or midterm from the moment solutions were posted on piazza or discussed in class
- Make sure you read and understand the solution before asking for a regrade
- Request for a regrade will result in the regrading of the entire homework, project or midterm

Academy integrity

- It is allowed to discuss homework problems before writing them down; however, WRITING IS INDIVIDUAL
  - if you look at another student's written or typed answers, or let another student look at your written or typed answers, that is considered cheating.
- Never have a copy of someone else's homework or program in your possession and never give your homework (or password) or program to someone else.
- NO CHEATHING WILL BE TOLERATED.
- ANY CHEATING WILL AUTOMATICALLY RESULT in F grade and report to the university administration

How to ask on Piazza

- Read slides, notes, homework or project description
- Use #hashtags (#lecture2, #project3, #hw1, etc.)
- Describe the problem clearly, using the right terms
- Add code in attached files
- Add output from compiler
- Add any other relevant information
- **Don't post solutions on piazza**
- **Anything that relates to solution post PRIVATELY**

Weather/Emergency

- In the event of a major campus emergency, course requirements, deadlines and grading percentages are subject to changes that may be necessitated by a revised semester calendar or other circumstances beyond the instructor's control.
- Monitor weather and piazza particularly if you don't live close to school.

Ethics

- We will talk much more about ethics and security later
- For now, follow these simple rules
  - Only develop and launch attacks against systems setup by us or yourself
  - Do not launch attacks against anyone else
- Attacking computers is a serious crime, punishable by huge fines and/or jail time

One last word …

- **No meetings will be accepted with the TA or instructor the day homework or projects are due, or the day of exam**
- Start early, plan carefully
- Develop your solution gradually, test gradually so you always have functionality for which you can receive a grade; YOUR CODE MUST WORK
- Do not wait to submit your code last minute
- Don't post solutions on piazza
- Don't cheat

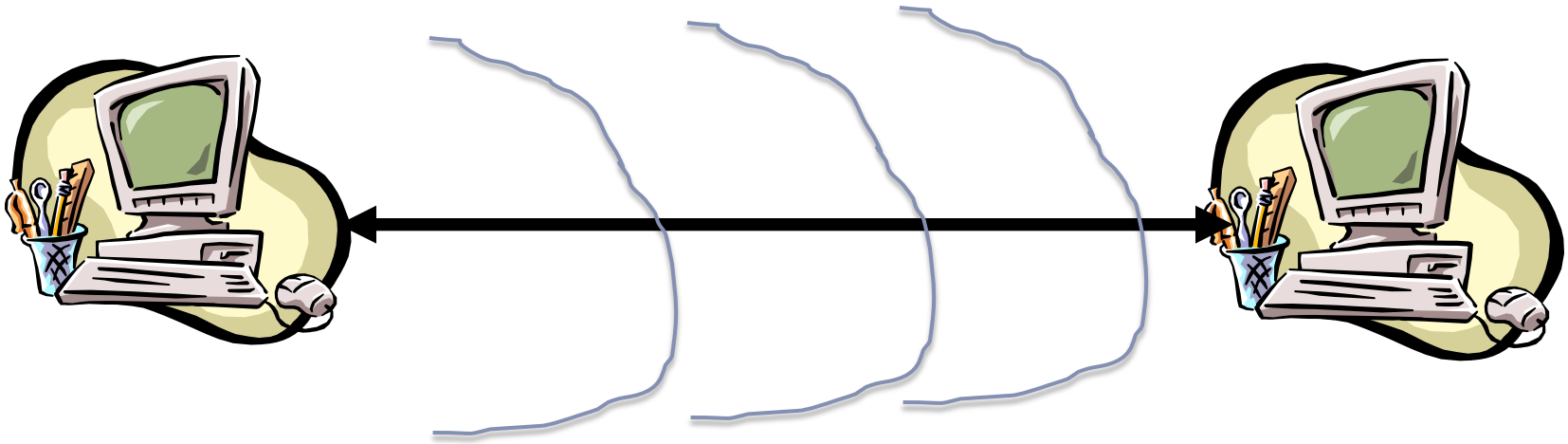
PIAZZA ACCOUNTS

- All communication is on piazza, make sure you get notifications and you check piazza constantly
- If you have not received a piazza notification email me c.nitarotaru@neu.edu

2: Security and privacy goals for network protocols

Lots of networks …

- Networks:
  - The Internet
  - Wifi
  - Cellular
  - Sensor
- Protocol: Defines rules of sending/receiving packets:
  - Format and type of the packets
  - Actions in response of receiving a certain type of packets …



Network protocols

- Network protocols do not exist in a vacum
- They facilitate interactions between
  - Different devices sometimes with different operating systems and different security policies
- Many protocols are implemented as part of the operating system and organized in stacks
- They depend on the physical medium where communication takes place
  - Particularly true for wireless communication
- They provide an entry/exit of information

- WHAT DOES SECURITY MEAN?

- WHAT DOES IS MEAN TO BE SECURE
  - For SOFTWARE, OS, **NETWORK PROTOCOLS**, etc
  - DESIGN or IMPLEMENTATION

Security is secondary

- Security is secondary to the interactions that make security necessary.
  - What protection/security mechanisms one has in the physical world?
  - Why the need for security mechanisms arises?

Robert H. Morris : The three golden rules to ensure computer security are: do not own a computer; do not power it on; and do not use it.

Security is not absolute

- Is your car secure?
- What does "secure" mean?
- Are you secure when you drive your car?

- Security is relative
  - to the kinds of loss one consider
    - security objectives/properties need to be stated
  - to the threats/adversaries under consideration
    - security is always under certain assumptions

Fundamental security goals

- Confidentiality
  - only those who are authorized to know can know
  - Example of attack breaking the goal for networking: packet sniffing
- Integrity (and authentication)
  - only modified by authorized parties and in authorized ways
  - Example of attack breaking the goal for networking: connection hijacking, packet injection
- Access control
  - only those who are authorized to perform a certain operation can do it
  - Example of attack breaking the goal for networking: only traffic from certain IPs is accepted
- Availability
  - service is available to those authorized to access it
  - Example of attack breaking the goal for networking: denial of service against a website

Examples

- ARP is not authenticated
  - APR spoofing (or ARP poisoning)
- Network packets pass by untrusted hosts
  - Packet sniffing
- TCP state can be easy to guess
  - TCP spoofing attack
- Open access
  - Vulnerable to DoS attacks
- DNS is not authenticated
  - DNS poisoning attacks

Tools for information security

- Cryptography
- Authentication and access control
- Hardware/software architecture for separation
- Processes and tools for developing more secure software
- Monitoring and analysis
- Recovery and response
- Physical protection

Attackers

- Interaction with data and protocol
  - Eavesdropping or actively participating in the protocol
- Resources
  - Computation, storage
  - Limited or unlimited
- Access to previously encrypted communication
  - Only encrypted information (ciphertext)
  - Pairs of message and encrypted version (plaintext, ciphertext)
- Interaction with the cipher algorithm
  - Choose or not for what message to have the encrypted version (chose ciphertext)

Interaction with data and protocol

- Passive: the attacker only monitors the communication. It threatens confidentiality.
  - Example: listen to the communication between Alice and Bob, and if it's encrypted try to decrypt it.
- Active: the attacker is actively involved in the protocol in deleting, adding or modifying data. It threatens all security services.
  - Example: Alice sends Bob a message: 'meet me today at 5', Carl intercepts the message and modifies it 'meet me tomorrow at 5', and then sends it to Bob.

Access control

- Policy specifying how entities can interact with resources
  - i.e., Who can access what?
  - Requires authentication and authorization
- Access control primitives

Principal	Users of a system
Subject	Entity that acts on behalf of principals
Object	Resource acted upon by subjects

Authentication

- Verification of identity claim made by a subject on behalf of a principal
- Involves examination of factors, or credentials
  - Something you have – e.g., a badge
  - Something you know – e.g., a password
  - Something you are – e.g., your fingerprint
- Desirable properties include being unforgeable, unguessable, and revocable

Authorization

- Authorization follows authentication
  - If asking what someone can do, you must know who they are
- Usually represented as a policy specification of what resources can be accessed by a given subject
  - Can also include the nature of the access

Types of access control

- Discretionary Access Control (DAC)
  - Owners of objects specify policy
- Mandatory Access Control (MAC)
  - Policy based on sensitivity levels – e.g., clearance
  - Owners do not specify their own policies
- Role-based Access Control (RBAC)
  - Central authority defines policy in terms of roles
  - Roles ≈ permission sets

Security principles

- Principle of weakest link
  - A system is as secure as its weakest link
- Principle of adequate protection
  - Maximize utility while limiting risk to an acceptable level within reasonable cost
- Principle of effectiveness
  - Controls must be efficient, easy to use, appropriate, and psychological acceptable
- Kerkoff' s principle
  - System design should be known, security relies on secrecy of secret key

- WHAT DOES ANONYMITY MEAN?

- WHAT DOES IT MEAN FOR A NETWORK PROTOCOL TO PROVIDE ANONYMOUS COMMUNICATION?

Anonymity

Anonymity (``without name'') means that a person is not identifiable within a set of subjects

- Unlinkability of action and identity
  - For example, sender and his email are no more related after adversary's observations than they were before
  - Who talks to whom
- Unobservability
  - Adversary cannot tell whether someone is using a particular system and/or protocol

Lack of privacy on public networks

- Internet is designed as a public network
  - Wi-Fi access points, network routers see all traffic that passes through them
- Routing information is public
  - IP packet headers identify source and destination
  - Even a passive observer can easily figure out who is talking to whom
- Encryption does not hide identities
  - Encryption hides payload, but not routing information

Anonymity goals

- Basic metrics:
  - Sender anonymity - who sends what
  - Receiver anonymity - who receives what
  - Unlinkability (relationship anonymity) - who talks to whom
- Providing sender anonymity and unlinkability are desirable enough for common Internet activities
- Goals:
  - The identities of the communicating parties should stay anonymous to the outside community
  - Even the parties in communication may not know each other's real identity

Types of adversary

- Passive/Active
  - Passive: eavesdrop traffic
  - Active: able to observe, delay, alter and drop messages
- Local/Global
  - Local: able to observe traffic to/form user's network link, within LAN
  - Global: able to observe effectively large amount or all network links, across LAN boundaries
- Internal/External
  - Internal: does participate in the anonymity system
  - External: does not participate in the system

Take home lessons

- Security and anonymity are relative
  - Goals are provided under specific adversarial models
- Goals
  - What the system/protocol promises
- Attacker model
  - How the attacker interacts with the system
  - What resources has available
- Boundaries
  - What is assumed about the context
  - Defines the boundaries of the secure system

3: Background: Security models.

Abstract security models

- Access control lists
- Capabilities
- Bell-LaPadula
- Biba Integrity
- Clark-Wilson
- Brewer-Nash
- Non-interference
- Information flow

Practical security models

- UNIX permissions
- Windows access control
- Java permissions
- Web (same-origin policy)
- Android permissions
- iOS (MAC model)

Access Control List (ACL)

- ⟨object, subject, operation⟩

- Authorization verified for each request by checking list of tuples
- Instantiation of access control matrices with update
- Used pervasively in filesystems and networks
  - "Users a, b, and c and read file x."
  - "Hosts a and b can listen on port x."
- Drawbacks?

Capabilities

- In this model, authorization is synonymous with possession of a capability
  - Capabilities represented as transferable, unforgeable tokens
- Many implementations
  - Hardware
  - Systems (EROS, Capsicum)
  - Languages (E, Caja, Joe-E)
- Drawbacks?

Bell-LaPadula (BLP)

- Concerned with enforcing confidentiality
- Subjects have clearances
  - e.g., Confidential, Secret, Top-Secret, TS/SCI
- Objects have classifications
- State-transition model specifies system evolution

Bell-LaPadula

- "No read up, no write down"

- Simple security property
  - A subject at a given level cannot read an object at a higher level
- ★-property (confinement)
  - A subject at a higher level cannot write to an object at a lower level
- Discretionary security property
  - Additional DAC – e.g., ACLs

Biba Integrity

- "No read down, no write up"

- Simple integrity axiom
  - Subjects at a higher level cannot read objects at a lower level
- ★-integrity axiom
  - Subjects at a lower level cannot write to objects at a higher level

Covert channels

- Access control is defined over "legitimate" channels
  - e.g., shared memory, pipes, sockets, files
- However, isolation in real systems is imperfect
- External observations can be used to create covert channels
  - Requires collusion with an insider
- Can be extremely difficult to detect
  - Difficulty is proportionate to channel bandwidth

Non-interference

- Any sequence of low inputs produces the same low outputs regardless of high inputs

- System modeled as machine with low (unprivileged) and high (privileged) inputs and outputs
- Property guarantees that regardless of high inputs, no externally observable effects occur in the low outputs
  - Guarantees no covert channels
  - Very strict and virtually unrealizable property

Information flow

- Traditional access control is coarse-grained
  - Access control specifies how information is released
- Information flow policies specify how information is propagated
  - Objects classified by levels
  - Policies denote allowable flows between subjects
- Distinction between explicit flows and implicit flows

Explicit/implicit flows

Given program variables h: high, l: low,

- Explicit information flows involve a direct transfer of information from high to low objects

```
l = h;          // Explicit flow of all bits of h
l = h % 2;      // Explicit flow of LSB of h
l ^= h >> 3;    // Explicit flow of high bits of h
```

- Implicit flows leak information from high to low objects via an indirect mechanism (e.g., control flow)

```
l = 0;
h = h % 2;
if (h == 1) {
        l = 1;
}
```

Information flow control

- Information flow control (IFC) makes it theoretically possible to verify non-interference
  - Within a given model of a system...
- However, realistic programs require declassification
  - i.e., most systems require flows from high to low
- Numerous implementations of IFC
  - Systems – e.g., Asbestos, Hi-Star
  - Languages – e.g., Jif, Sif

Side channels

- Side channels result from inadvertent information leakage
  - Timing – e.g., password recovery by timing keystrokes
  - Power – e.g., crypto key recovery by power fluctuations
  - RF emissions – e.g., video signal recovery from video cable EM leakage
  - Virtually any shared resource can be used
- Countermeasures?
  - Remove access to shared resource
  - Introduce noise (chaff) or blind the resource

Take home lessons

- Different theoretical and practical models for security
- Discretionary Access Control vs Mandatory Access Control
- In practice it is impossible to eliminate
  - Covert channels
  - Side-channels

4: Background: Cryptographic building blocks

Readings for this section

- Required readings:
  - Cryptography on Wikipedia
- Interesting reading
  - The Code Book by Simon Singh

Symmetric-key encryption

- A symmetric-key encryption scheme consists of three algorithms
  - Gen: the key generation algorithm
    - The algorithm must be probabilistic/randomized
    - Output: a key k
  - Enc: the encryption algorithm
    - Input: key k, plaintext m
    - Output: ciphertext c := Enck(m)
  - Dec: the decryption algorithm
    - Input: key k, ciphertext c
    - Output: plaintext m := Deck(m)

Requirement: $\forall k \, \forall m \;\; [\, \mathbf{Dec}_k(\mathbf{Enc}_k(m)) = \mathrm{m} \,]$

Secret key (symmetric) building blocks

- Confidentiality
  - Stream ciphers (uses PRNG)
  - Block ciphers with encryption modes
- Integrity
  - Cryptographic hash functions
  - Message authentication code (keyed hash functions)
- Limitation: sender and receiver must share the same key
  - Needs secure channel for key distribution
  - Impossible for two parties having no prior relationship
  - Needs many keys for n parties to communicate

Block ciphers

- An n-bit plaintext is encrypted to an n-bit ciphertext
  - P : {0,1}n
  - C : {0,1}n
  - K : {0,1}s
  - E: K ×P → C : E_k: a permutation on {0,1} n
  - D: K ×C → P : D_k is E_{k-1}
  - Block size: n
  - Key size: s

Data Encryption Standard (DES)

- Designed by IBM, with modifications proposed by the National Security Agency
- US national standard from 1977 to 2001
- Block size is 64 bits;
- Key size is 56 bits
- Has 16 rounds
- Designed mostly for hardware implementations
  - Software implementation is somewhat slow
- Considered insecure now
  - vulnerable to brute-force attacks
- 2DES insecure too, 3DES still used

AES

- Designed to be efficient in both hardware and software across a variety of platforms
- Block size: 128 bits
- Variable key size: 128, 192, or 256 bits.
- No known weaknesses
- De facto standard

Block Cipher Encryption Modes: ECB

- Message is broken into independent blocks;
- Electronic Code Book (ECB): each block encrypted separately.
- **Encryption: $c_i = E_k(x_i)$**
- **Decrytion: $x_i = D_k(c_i)$**

Properties of ECB

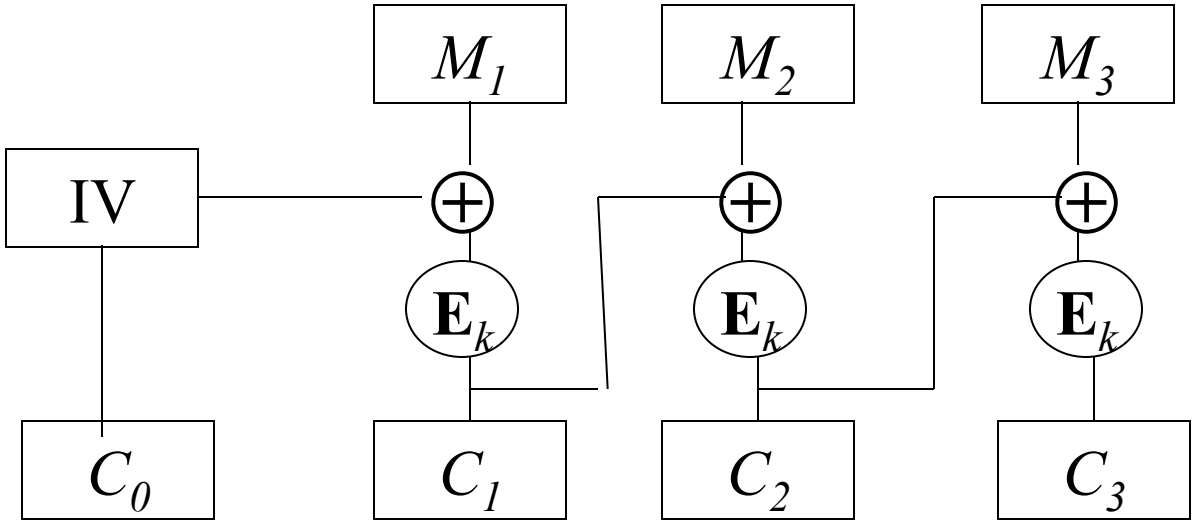
- Deterministic:
  - the same data block gets encrypted the same way,
    - reveals patterns of data when a data block repeats
  - when the same key is used, the same message is encrypted the same way
- Usage: not recommended to encrypt more than one block of data

DES Encryption Modes: CBC

- Cipher Block Chaining (CBC):
  - Uses a random Initial Vector (IV)
  - Next input depends upon previous output

Encryption: $C_i = E_k(M_i \oplus C_{i-1})$, **with** $C_0 = IV$

Decryption: $M_i = C_{i-1} \oplus D_k(C_i)$, **with** $C_0 = IV$

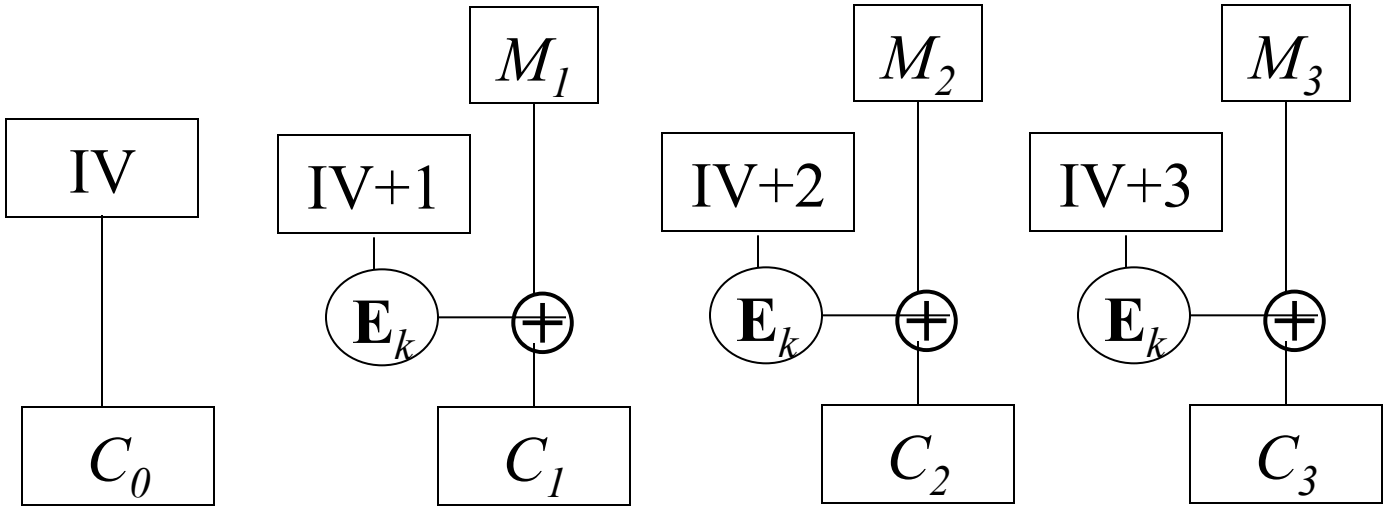


Properties of CBC

- Randomized encryption: repeated text gets mapped to different encrypted data.
  - can be proven to provide IND-CPA assuming that the block cipher is secure (i.e., it is a Pseudo Random Permutation (PRP)) and that IV's are randomly chosen and the IV space is large enough (at least 64 bits)
- Each ciphertext block depends on all preceding plaintext blocks.
- Usage: chooses **random** IV and protects the **integrity** of IV
  - The IV is not secret (it is part of ciphertext)
  - The adversary cannot control the IV

Encryption modes: CTR

- Counter Mode (CTR): Defines a stream cipher using a block cipher
  - Uses a random IV, known as the counter
  - Encryption: C_0=IV, $C_i = M_i \oplus E_k[IV+i]$
  - Decryption: IV=C_0, $M_i = C_i \oplus E_k[IV+i]$



Properties of CTR

- Gives a stream cipher from a block cipher
- Randomized encryption:
  - when starting counter is chosen randomly
- Random Access: encryption and decryption of a block can be done in random order, very useful for hard-disk encryption.
  - E.g., when one block changes, re-encryption only needs to encrypt that block. In CBC, all later blocks also need to change

Hash Functions

- A hash function maps a message of an arbitrary length to a m-bit output
  - output known as the fingerprint or the message digest
- What is an example of hash functions?
  - Give a hash function that maps Strings to integers in [0,2^{32}-1]
- Cryptographic hash functions are hash functions with additional security requirements

Security requirements for cryptographic hash functions

Given a function $h:X \rightarrow Y$, then we say that h is

- preimage resistant (one-way):

  if given $y \in Y$ it is computationally infeasible to find a value $x \in X$ s.t. $h(x) = y$

- 2-nd preimage resistant (weak collision resistant):

  if given $x \in X$ it is computationally infeasible to find a value $x' \in X$, s.t. $x' \neq x$ and $h(x') = h(x)$

- collision resistant (strong collision resistant):

  if it is computationally infeasible to find two distinct values $x', x \in X$, s.t. $h(x') = h(x)$

Using hash functions for message integrity

- Method 1: Uses a hash function h, assuming an authentic (adversary cannot modify) channel for short messages
  - Transmit a message M over the normal (insecure) channel
  - Transmit the message digest h(M) over the secure channel
  - When receiver receives both M' and h, how does the receiver check to make sure the message has not been modified?

- This is insecure. How to attack it?
- A hash function is a many-to-one function, so collisions can happen.

Choosing the length of Hash outputs

- The Weakest Link Principle:
  - A system is only as secure as its weakest link.
- Hence all links in a system should have similar levels of security.
- Because of the birthday attack, the length of hash outputs in general should double the key length of block ciphers
  - SHA-224 matches the 112-bit strength of triple-DES (encryption 3 times using DES)
  - SHA-256, SHA-384, SHA-512 match the new key lengths (128,192,256) in AES

HMAC: Constructing MAC from cryptographic hash functions

- K+ is the key padded (with 0) to B bytes, the input block size of the hash function
- ipad = the byte 0x36 repeated B times
- opad = the byte 0x5C repeated B times.

$$HMAC_K[M] = Hash[(K^+ \oplus opad) \,||\, Hash[(K^+ \oplus ipad)||M)]]$$

HMAC Security: If used with a secure hash function and according to the specification (key size, and use correct output), no known practical attacks

SHA-1

- Hash function used for a long time and subjected to numerous attacks
- Brute force attack is harder (160 vs 128 bits for MD5)
- Wang, Yin, and Yu (2005) found ways to find collisions using no more than 269 hash evaluations
- Wang, Yao and Yao (2005) found collisions using no more than 263 hash evaluations
- NIST made a request for the design of a new hash function; Replaced by SHA-3

SHA-3

- NIST had an ongoing competition for SHA-3, the next generation of standard hash algorithms
  - 2007: Request for submissions of new hash functions
  - 2008: Submissions deadline. Received 64 entries. Announced first-round selections of 51 candidates.
  - 2009: After First SHA-3 candidate conference in Feb, announced 14 Second Round Candidates in July.
  - 2010: After one year public review of the algorithms, hold second SHA-3 candidate conference in Aug. Announced 5 Third-round candidates in Dec.
  - 2011: Public comment for final round
- 2012: October 2, NIST selected SHA3
  - Keccak (pronounced "catch-ack") created by Guido Bertoni, Joan Daemen and Gilles Van Assche, Michaël Peeters

Public key encryption

- Each party has a pair (K, K-1) of keys:
  - K is the public key, and used for encryption
  - K-1 is the private key, and used for decryption
  - Satisfies DK-1[EK[M]] = M
- Knowing the public-key K, it is computationally infeasible to compute the private key K-1
  - How to check (K,K-1) is a pair?
  - Offers only computational security. Secure PK Encryption impossible when P=NP, as deriving K-1 from K is in NP.
- The public-key K may be made publicly available, e.g., in a publicly available directory
  - Many can encrypt, only one can decrypt

Public key building blocks

- Confidentiality
  - ElGamal
  - RSA
- Non-repudiation with/wo Integrity
  - Digital signatures
- Limitation: sender and receiver must obtain the public key in a ``secure way''
  - Need for PKI and certificate distribution

ElGamal encryption

- Public key <g, p, $h=g^a \bmod p$>
- Private key is a
- To encrypt: chooses random b, computes
  $C=[g^b \bmod p, g^{ab} * M \bmod p]$.
  - Idea: for each M, sender and receiver establish a shared secret g^{ab} via the DH protocol. The value g^{ab} hides the message M by multiplying it.
- To decrypt $C=[c_1,c_2]$, computes M where
  - $((c_1^a \bmod p) * M) \bmod p = c_2$.
    - To find M for $x * M \bmod p = c_2$, compute z s.t. $x*z \bmod p = 1$, and then $M = C_2*z \bmod p$

RSA

- Invented in 1978 by Ron Rivest, Adi Shamir and Leonard Adleman
  - Published as R L Rivest, A Shamir, L Adleman, "On Digital Signatures and Public Key Cryptosystems", Communications of the ACM, vol 21 no 2, pp120-126, Feb 1978
- Security relies on the difficulty of factoring large composite numbers
- Essentially the same algorithm was discovered in 1973 by Clifford Cocks, who works for the British intelligence

RSA key generation

1. Select 2 large prime numbers of about the same size, p and q
   Typically each p, q has between 512 and 2048 bits
2. Compute n = pq, and $\Phi(n) = (q-1)(p-1)$
3. Select e, $1<e<\Phi(n)$, s.t. $\gcd(e, \Phi(n)) = 1$
   Typically e=3 or e=65537
4. Compute d, $1< d<\Phi(n)$ s.t. $ed \equiv 1 \bmod \Phi(n)$
   Knowing $\Phi(n)$, d easy to compute.

Public key: (e, n)
Private key: d

RSA encryption/decryption

Encryption

Given a message M, $0 < M < n$ $M \in Z_n - \{0\}$

use public key (e, n)

compute $C = M^e \bmod n$ $C \in Z_n - \{0\}$

Decryption

Given a ciphertext C, use private key (d)

Compute $C^d \bmod n = (M^e \bmod n)^d \bmod n = M^{ed} \bmod n = M$

RSA example

- p = 11, q = 7, n = 77, $\Phi(n) = 60$
- d = 13, e = 37 (ed = 481; ed mod 60 = 1)
- Let M = 15. Then $C \equiv M^e \bmod n$
  - $C \equiv 15^{37} \pmod{77} = 71$
- $M \equiv C^d \bmod n$
  - $M \equiv 71^{13} \pmod{77} = 15$

Non-repudiation

- Nonrepudiation is the assurance that someone cannot deny something. Typically, nonrepudiation refers to the ability to ensure that a party to a contract or a communication cannot deny the authenticity of their signature on a document or the sending of a message that they originated.
- Can one deny a signature one has made?
- Does email provide non-repudiation?

Key Agreement: Diffie-Hellman Protocol

- Key agreement protocol, both A and B contribute to the key
- Setup Z_n, n prime and g generator, n and g public.

$g^a \bmod n$

$g^b \bmod n$

Pick random, secret a
Compute and send $g^a \bmod n$

$K = (g^b \bmod n)^a = g^{ab} \bmod n$

Pick random, secret b
Compute and send $g^b \bmod n$

$K = (g^a \bmod n)^b = g^{ab} \bmod n$

Diffie-Hellman

- Example: Let p=11, g=2, then

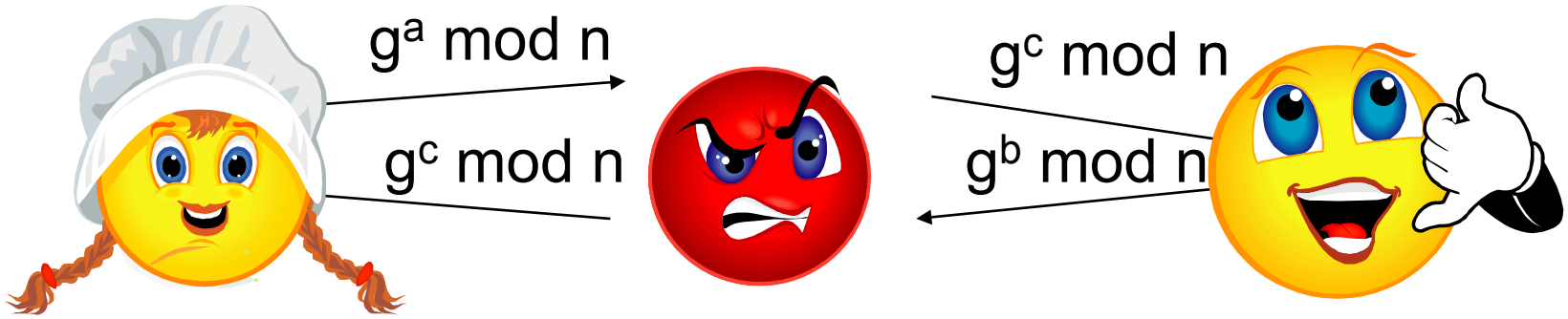
a	1	2	3	4	5	6	7	8	9	10	11
g^a	2	4	8	16	32	64	128	256	512	1024	2048
g^a mod p	2	4	8	5	10	9	7	3	6	1	2

A chooses 4, B chooses 3, then shared secret is

$$(2^3)^4 = (2^4)^3 = 2^{12} = 4 \pmod{11}$$

Adversaries sees $2^3=8$ and $2^4=5$, needs to solve one of $2^x=8$ and $2^y=5$ to figure out the shared secret.

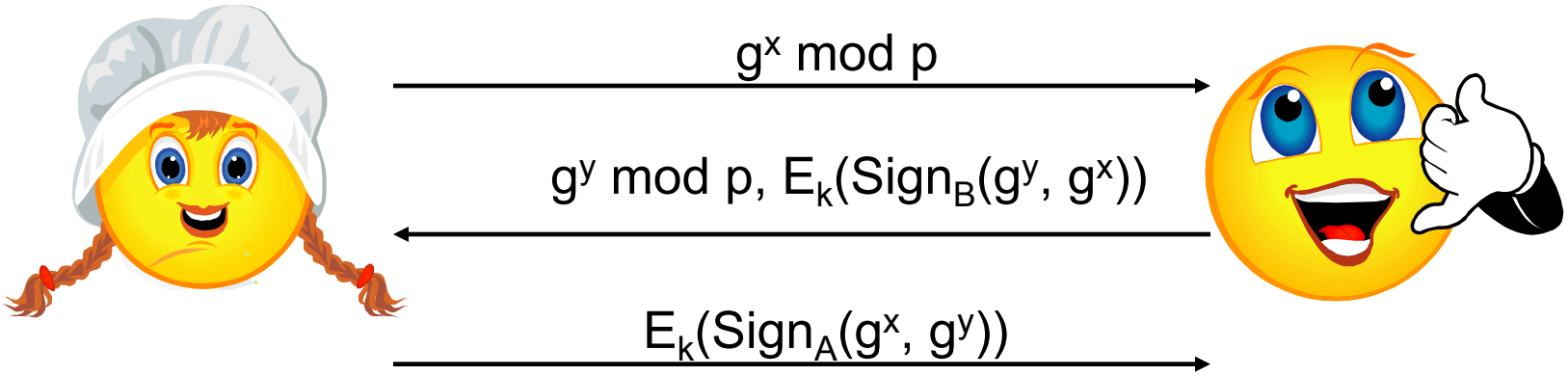
Man-in the Middle-Attack Against Unauthenticated DH



Alice computes $g^{ac} \bmod n$ and Bob computes $g^{bc} \bmod n$!!!
Attacker can compute both keys.

Station-to-Station (STS)

Provides mutual entity authentication



Take home lessons

	Secret Key Setting	Public Key Setting
Secrecy / Confidentiality	Stream ciphers Block ciphers + encryption modes	Public key encryption: RSA, El Gamal, etc.
Authenticity / Integrity	Message Authentication Code	Digital Signatures: RSA, DSA, etc.