CMPE 220

Class 17 – System Security

Protection

- An operating system must protect users and processes from each other's activities.
  - Mechanisms to control the access by programs, processes, or users to the computer system resources.
- Each resource object has a unique name and can be accessed through

  a well-defined set of operations.
  - Ensure that each object is accessed correctly and only by those processes that are allowed to do so.

San José State University

Principle of Least Authority (POLA)

- A guiding principle of protection.
- Give programs, users and systems just enough privileges to perform their tasks.
- Limit damage if an entity has a bug or gets abused.
- "Need to know"
  - At any time, a process should be able to access only those resources that it currently requires to complete its task.

Protection Granularity

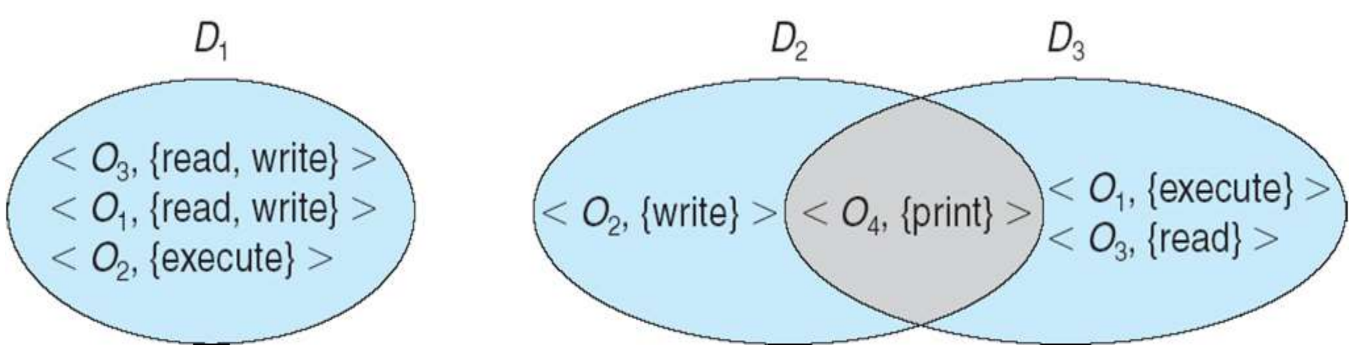
- <u>Rough-grained</u> privilege management is easier and simpler.
  - Principle of least authority done in large chunks.
  - Example: Traditional UNIX processes either have abilities of the associated user or of the root user.

- <u>Fine-grained</u> management is more complex and more overhead, but more protective.
  - Examples:
    - File access control lists (ACLs)
    - Role-based access control (RBAC)

Protection Domains

- A domain is a set of resource objects and their allowable operations.
  - A set of access rights as <object-name, rights-set> pairs
  - Example: <file *F*, {read, write}>

- Users, processes, and program procedures can be in domains.



UNIX Domains

- In UNIX, a process's domain is defined by its UID (user ID) and GID (group ID).

- A (UID, GID) combination determines:
  - A complete list of all the accessible resource objects.
  - Whether they can be read, written, or executed.

UNIX Domains, *cont'd*

- Two processes with the same (UID, GID) combination have access to the same set of objects.
- Processes with different (UID, GID) combinations will have access to different but possibly overlapping sets of objects.

UNIX Domain Switching

- A system call causes a domain switch
  from the user's domain to that of the kernel.

- When a process executes a file with the SETUID or SETGID bit on,
  it acquires the UID or GID of the file owner.
  - Get a different set of access rights.
  - The UID and GID are reset when the execution completes.

San José State
UNIVERSITY

UNIX Domain Switching, *cont'd*

- Domain switch accomplished via passwords.
  - The **`su`** command temporarily switches to another user's domain when the other domain's password is provided.

- Domain switching via commands
  - The **`sudo`** command prefix executes a specified command in another domain.

The Access Matrix

object / domain	F_1	F_2	F_3	printer
D_1	read		read	
D_2				print
D_3		read	execute	
D_4	read write		read write	

- Rows represent domains
- Columns represent objects
- Access(*i, j*) is the set of operations that a process executing in Domain$_i$ can invoke on Object$_j$

Operating System Concepts, 10th edition
by Abraham Silberschatz, Greg Gagne, and Peter B. Galvin
Wiley, 2018, ISBN 978-1119456339

The Access Matrix, *cont'd*

- Domains are also objects.
  - Allow switching operations between domains.

domain \ object	F_1	F_2	F_3	laser printer	D_1	D_2	D_3	D_4
D_1	read		read			switch		
D_2				print			switch	switch
D_3		read	execute					
D_4	read write		read write		switch			

- The access matrix is generally very sparse.
  - Can be implemented in other more efficient ways.

Operating System Concepts, 10th edition
by Abraham Silberschatz, Greg Gagne, and Peter B. Galvin
Wiley, 2018, ISBN 978-1119456339

The Access Matrix, *cont'd*

- Separate mechanism from policy.

- Mechanism
  - The OS provides access matrix + rules.
  - The OS ensures that the matrix is only manipulated by authorized agents and that rules are strictly enforced.

- Policy
  - User dictates policy.
  - Who can access what object and in what mode.

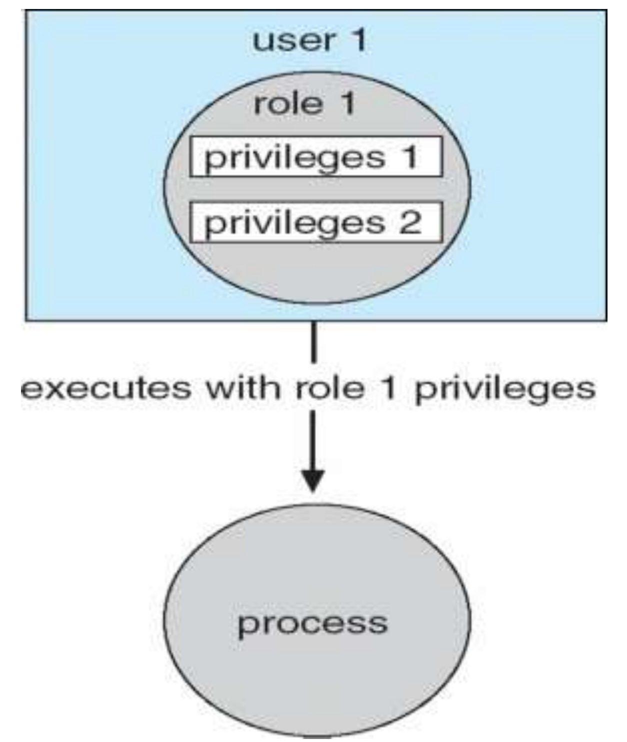
San José State University

Role-Based Access Control (RBAC)

- Implements the principle of least authority.

- Users are assigned roles that grant access to privileges and programs

  - Enable a role via a password to gain its privileges.



Operating System Concepts, 10th edition
by Abraham Silberschatz, Greg Gagne, and Peter B. Galvin
Wiley, 2018, ISBN 978-1119456339

Program-Level Protection

- Programming languages can have built-in protection mechanisms.

- Allow the high-level description of policies for the allocation and use of resources.

- Provide software for protection enforcement when automatic hardware-supported checking is unavailable.

San José State University

Security

- Protection mechanisms protect against internal problems.

- Security measures protect against external threats.

Security Violations

- Breach of confidentiality
  - Unauthorized reading of data.
- Breach of integrity
  - Unauthorized modification of data.
- Breach of availability
  - Unauthorized destruction of data.

Security Violations, *cont'd*

- Theft of service
  - Unauthorized use of resources.
- Denial of service (DOS)
  - Prevention of legitimate use.

San José State UNIVERSITY

Security Violation Methods

- Masquerading (breach authentication)
  - Pretend to be an authorized user to escalate privileges.
- Replay attack
  - With or without message modification.
- Session hijacking
  - Intercept an already-established session to bypass authentication.

San José State University

Security Violation Methods

- Man-in-the-middle attack
  - An intruder sits in data flow to masquerade as the sender in order to fool the receiver, and vice versa.

San José State University

The Bad Guys

- “Script kiddies”
  - “Hackers” who run malicious scripts
    that are shared among the hacker communities.
  - Can be thwarted by “honey pots”
    - Fake data at a site designed to lure hackers.
  - **Best defense: up-to-date software**

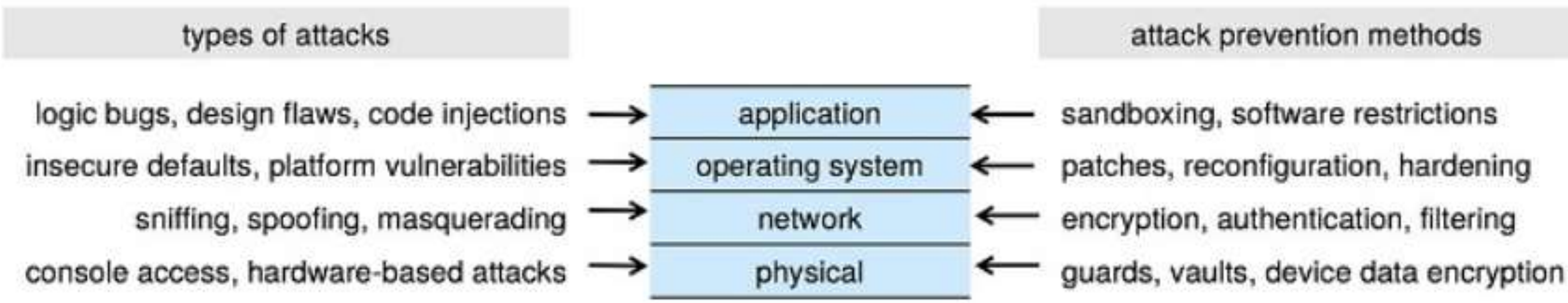
- Corporate thieves
  - Steal confidential data from competitors.

- Hostile (or friendly) governments
  - Snooping and monitoring
  - Spying

Layers of Security

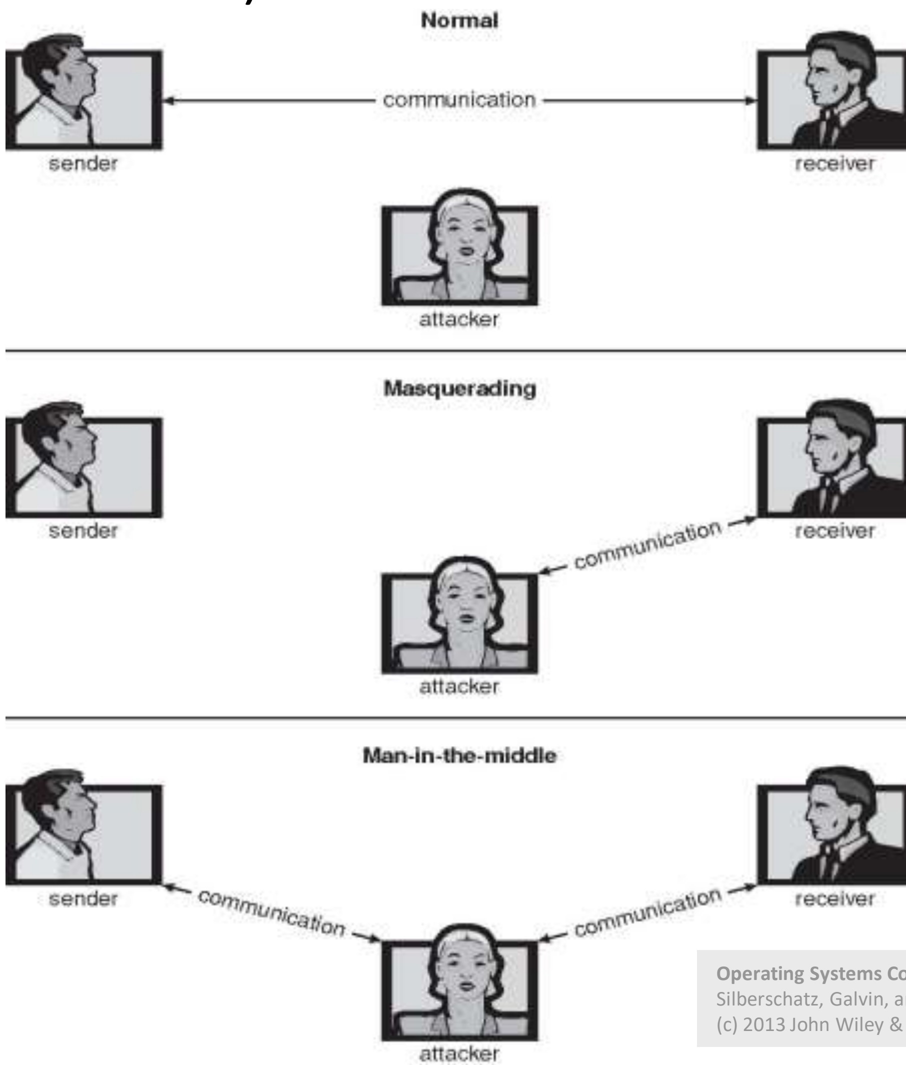
- It is impossible to have <u>absolute</u> security.
- Make the <u>cost</u> to the perpetrator sufficiently high to deter most intruders.
- Security is as strong as the weakest link in the chain.
- But can too much security be a problem?

Layers of Security, *cont'd*



Operating System Concepts, 10th edition
by Abraham Silberschatz, Greg Gagne, and Peter B. Galvin
Wiley, 2018, ISBN 978-1119456339

Man (or Woman) in the Middle Attack



Operating Systems Concepts, 9th edition
Silberschatz, Galvin, and Gagne

Trojan Horse Attack

- A program written by one user can execute in another user’s environment.
  - The program gains the other user’s access rights.
  - The program misuses those rights.
- A long UNIX path names exposes each directory on the path.
- A path that includes “.” when used in another user’s directory can give a program access to the other user’s home directory.

Trojan Horse Attack, *cont'd*

- Examples:
  - spyware
  - pop-up browser windows
  - browser plug-ins
  - covert channels

- Up to 80% of spam is delivered by spyware-infected systems.

Trap Door Attack

- Specific user identifier or password that circumvents normal security procedures.

Logic Bomb

- A program initiates a security incident under certain circumstances.
- Developed by a disgruntled programmer.
  - Must enter a password daily to prevent the bomb from going off.
- If the programmer is fired, the bomb explodes.
- Must hire the programmer back as an expensive consultant to “solve” the problem.

San José State University

Stack and Buffer Overflow

- Exploit a bug in a program to gain unauthorized user or privilege escalation.
  - Overflow either the stack or memory buffers.
  - Fail to check bounds on inputs or arguments.
- Write past the arguments on the stack into the return address on stack.
  - When routine returns from a function call,
    it returns to a hacked address.

Viruses

- A malicious code fragment embedded in a legitimate program.
- Self-replicating, designed to infect other computers.
- Very specific to CPU architecture, operating system, applications.
- Usually borne via email or as a macro.

San José State University

Categories of Viruses

- Parasitic file
- Boot
- Macro
- Source code
- Polymorphic
  - Avoids having a virus signature
- Encrypted
  - Encrypted to avoid detection.
  - Decrypts to execute.

San José State University

Categories of Viruses, *cont'd*

- Stealth
  - Modifies parts of the system that can be used to detect it.
- Tunneling
  - Installs in the interrupt-handler chain or in device drivers.
- Multipartite
  - Infect multiple parts of a system.
- Armored
  - Hard for antivirus researchers to detect.

Ransomware

- A virus that encrypts important data on a user's computer system.
  - The villain demands payment (often in bitcoins) for the decryption key.
- Threaten to post stolen private data on the web.
  - The villain demands payment (often in bitcoins) or the data will be made public.

Keystroke Logger Virus

- A virus that intercepts keystrokes.
- Records passwords, etc.
- Sends confidential information
  to a malicious recipient.

Port Scanning

- Automated attempt to connect
  to a range of ports on one IP address
  or on a range of IP addresses.

San José State University

Denial of Service

- Overload the targeted computer to prevent it from doing any useful work.

- A distributed denial-of-service (DDOS) comes from multiple sites at once.
  - "Ping" of death.

- Consider traffic to a web site.
  - How can you tell the difference between being a target and being really popular?
  - Accidental: Students writing bad `fork()` code.

San José State University

Design Principles for Security

- The system design should be public.
- The default should be no access.
- Check for current authority.
- Give each process the <u>least authority possible</u>.

- The protection mechanism should be simple, uniform, and built into the lowest layers of the system.

- The scheme chosen must be psychologically acceptable.

San José State University

User Authentication: Passwords

- Passwords are often easy to guess.
- A classic research study compiled a list of likely passwords.
  - first and last names
  - street and city names
  - words from a moderate-sized dictionary
  - license plate numbers
  - short strings of random numbers
  - Discovered that over 86% of passwords then in use were in their list.

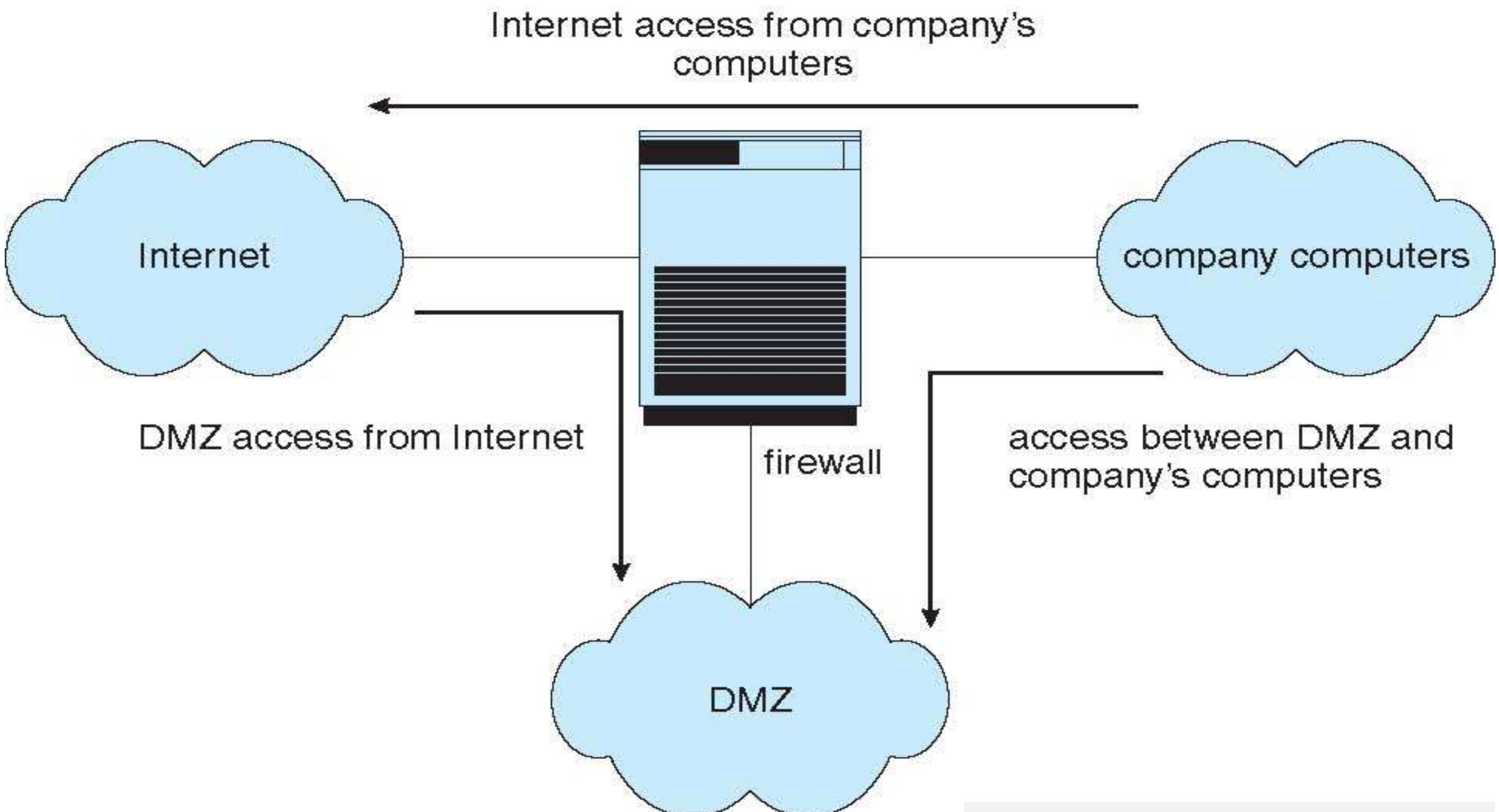
San José State University

Best Passwords

- ~~Upper case, lower case, digit, symbol~~
  - Hard to remember, so users write it down
  - Very difficult to enter on smartphones

- A long text string
  - Easy to remember, easy to type

Security Firewalls



Operating System Concepts, 10th edition
by Abraham Silberschatz, Greg Gagne, and Peter B. Galvin
Wiley, 2018, ISBN 978-1119456339

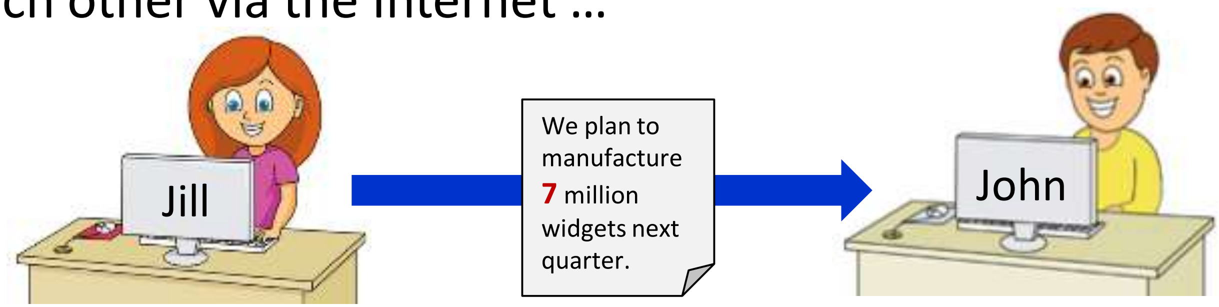
San José State University

The Biggest Risk to Computer Security

- PEOPLE!
  - Nefarious
  - Dumb

San José State University

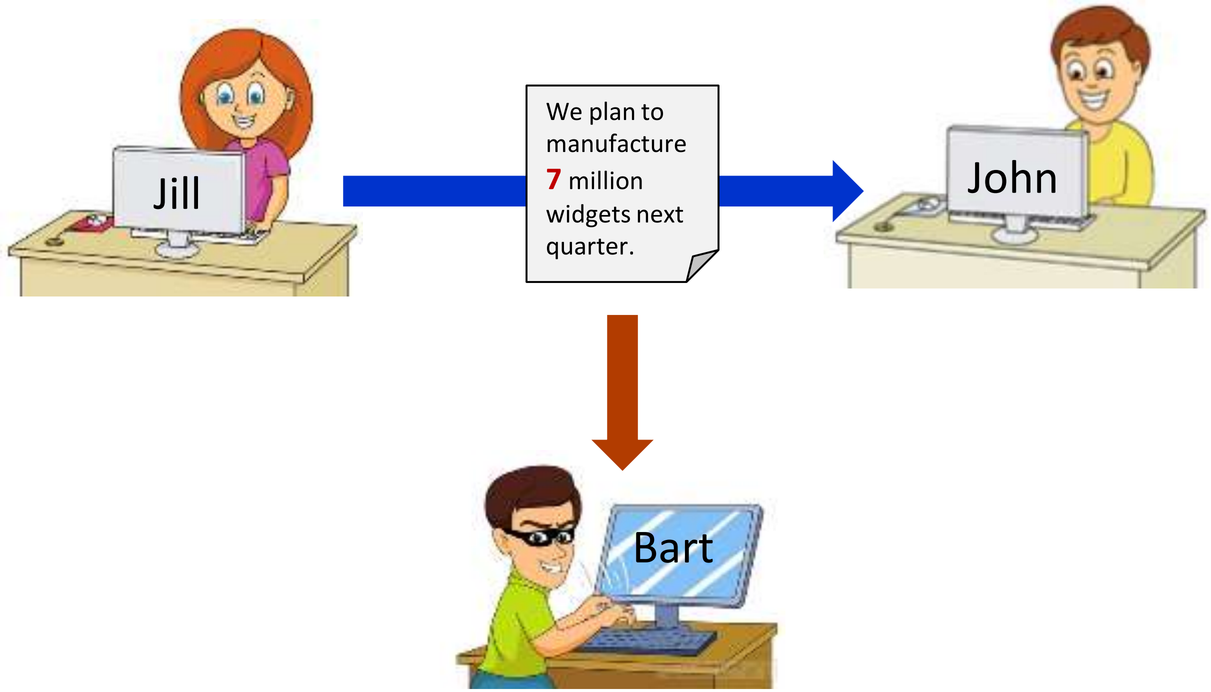
Nontechnical Security Lecture

- Sending data such as email messages to each other via the Internet ...
- 

  ... is like sending <u>postcards</u> via the U.S. mail system.

- Anyone can read the message along the way!

Security, *cont'd*

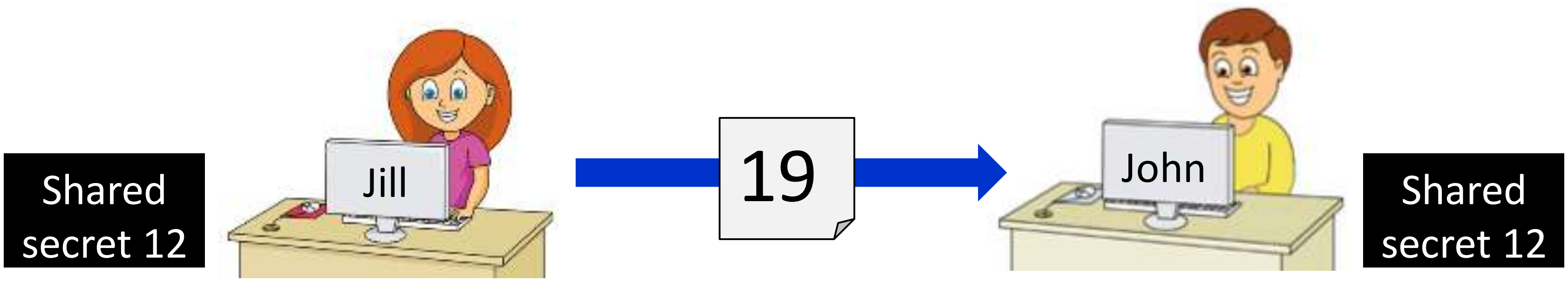


- How can we keep the <u>nefarious Bart</u> from reading confidential messages that Jill and John are sending each other?

San José State University

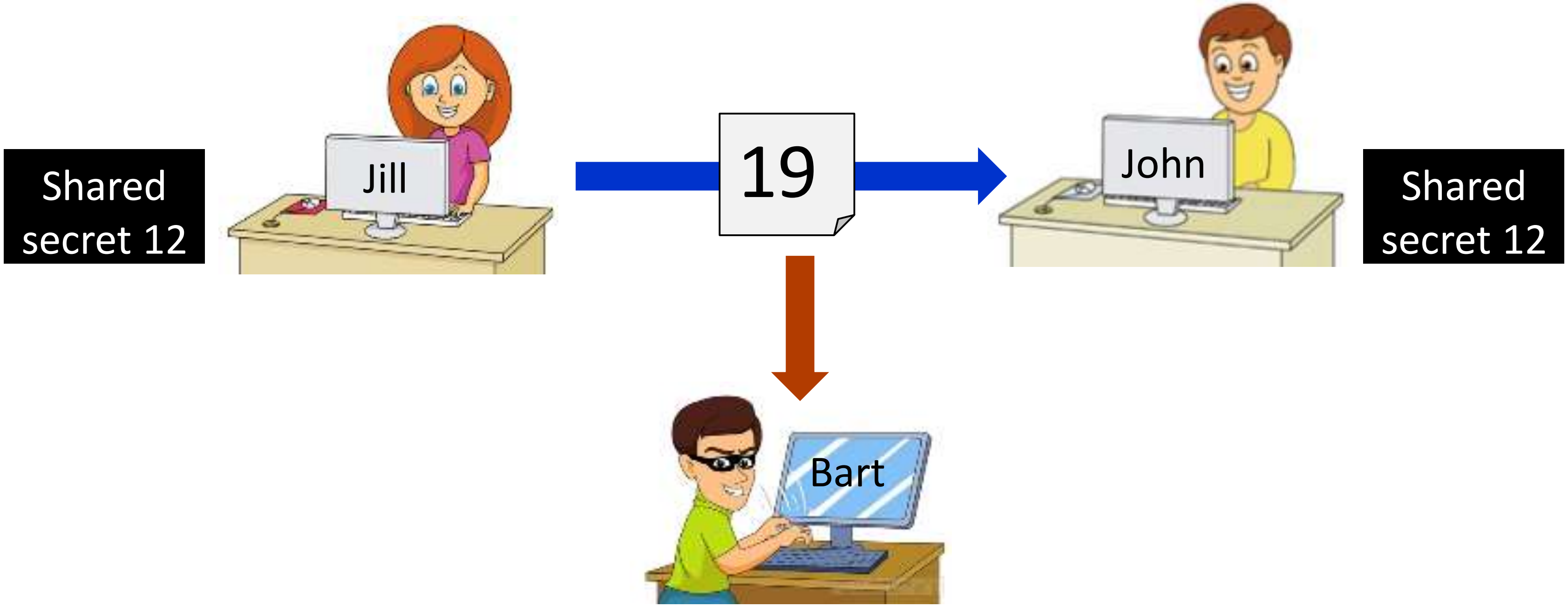
The Shared Secret

- Jill needs to send a message containing the confidential data 7 to John.



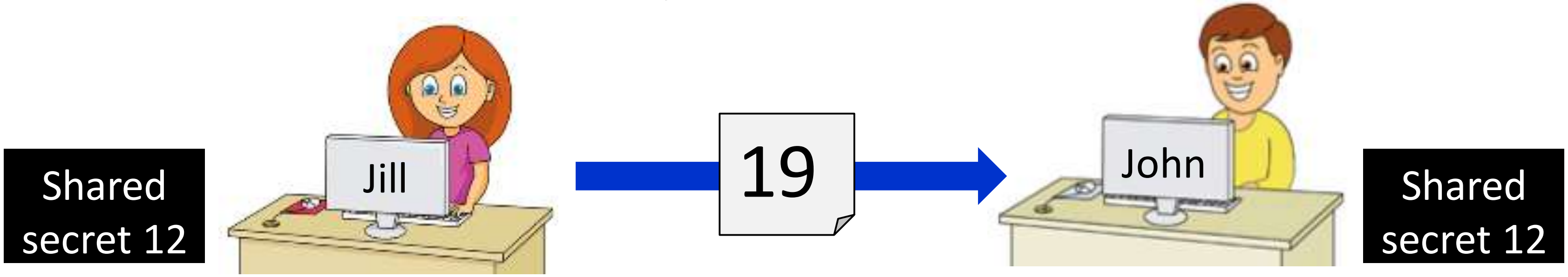
- John and Jill can agree ahead of time to a shared secret – the number 12.
- Then Jill can encrypt the data by adding 12 to the confidential data 7.
- John decrypts the data by subtracting 12.

The Shared Secret, *cont'd*



- Because Bart doesn't know the shared secret 12,
  he won't be able to decrypt the message and obtain the confidential data 7.

The Shared Secret, *cont'd*



- But this shared secret solution has problems.
  - Jill and John must arrange beforehand to share the secret 12.
  - What if Jill doesn't already know John?
  - What if Jill wants to send the confidential data to all her vice presidents at the same time?

How can Jill and her recipients share a secret?

San José State University

Public Key Cryptography, *cont'd*

- How can Jill and her recipients share a secret number in order to encrypt the confidential data?

- A security scheme called public key cryptography was invented just for this purpose.

- In this simplified introduction, let's pretend that multiplication is a one-way operation.
  - Once you've multiplied two numbers, say 4x5=20, you can't recover the original numbers by dividing.
  - In other words, you can't do 20÷4=5 or 20÷5=4

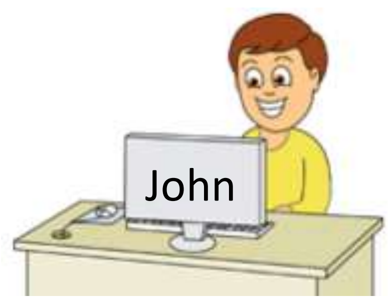
San José State University

Public Key Cryptography, *cont'd*

- Jill chooses a private key.
  - Let's suppose Jill chooses 10.

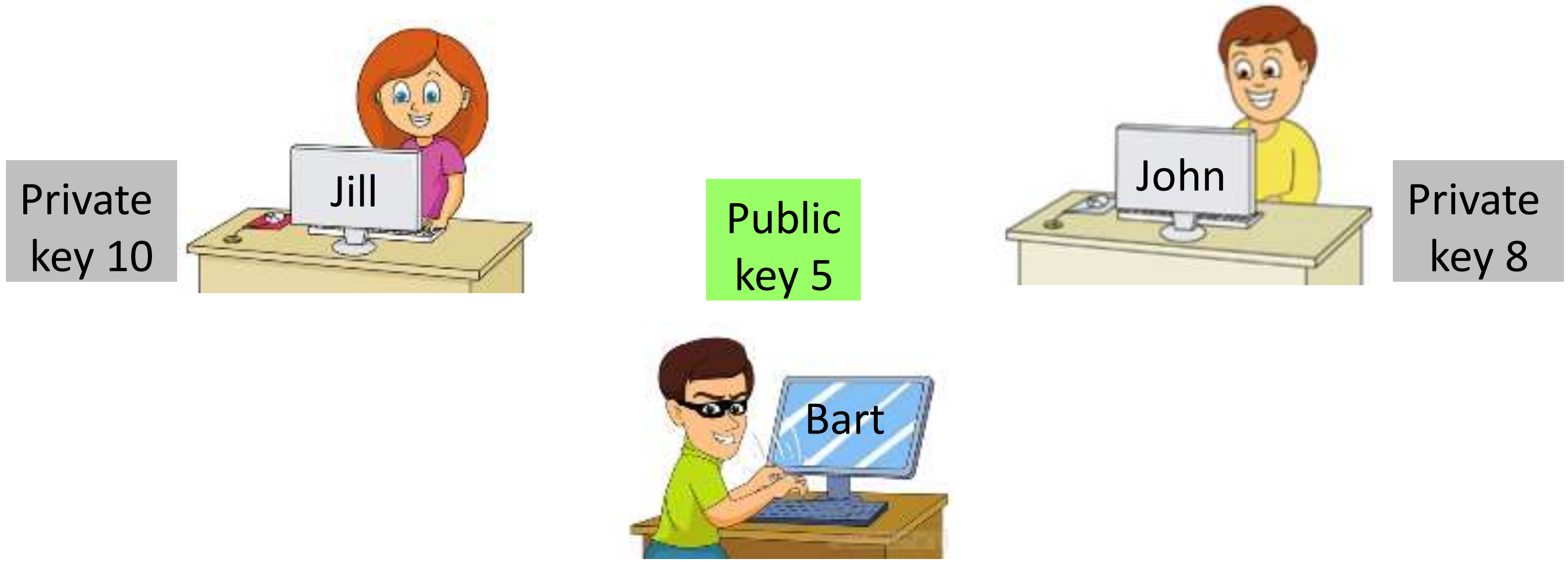
- Each person to whom Jill wants to send confidential data also chooses a private key.
  - Let's suppose John chooses 8.

Private key 10



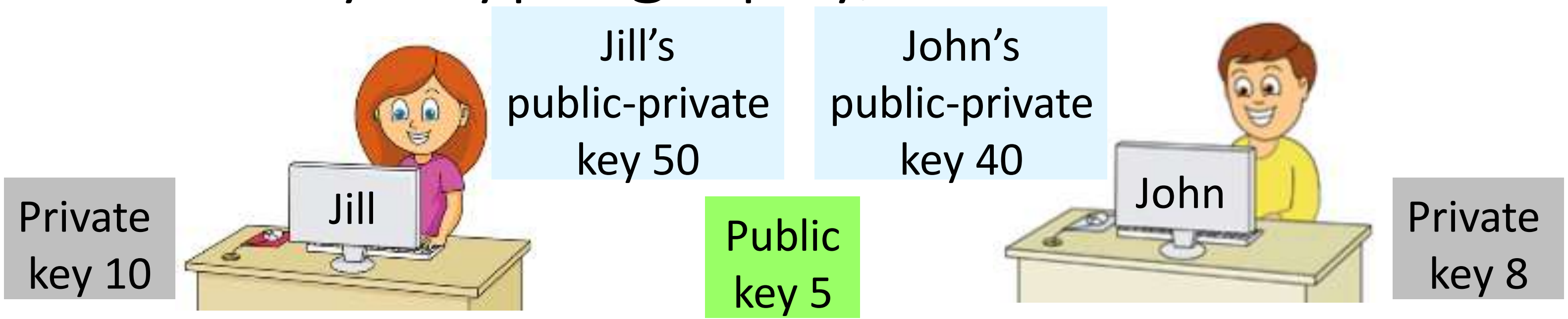
Private key 8

Public Key Cryptography, *cont'd*



- Now Jill announces a public key.
  - Let's suppose the public key is 5.
- Everyone can see the public key.
  - Including the nefarious Bart.

Public Key Cryptography, *cont’d*



- Now Jill can create her public-private key.
  - She multiplies her private key by the public key: 10x5=50.

- John creates his public-private key.
  - He multiplies his private key by the public key: 8x5=40.

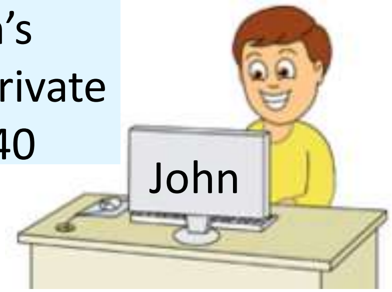
Public Key Cryptography, *cont'd*

Private key 10

Jill's public-private key 50

Public key 5

John's public-private key 40



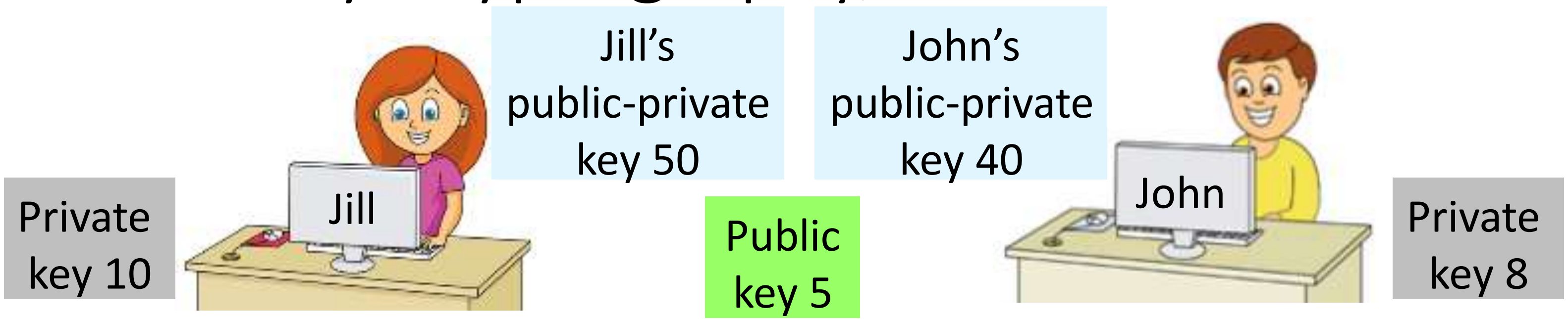
Private key 8

Remember that we're pretending that multiplication is a one-way operation.

- We <u>cannot</u> discover Jill's private key 10 by dividing her public-private key 50 by the public key 5.

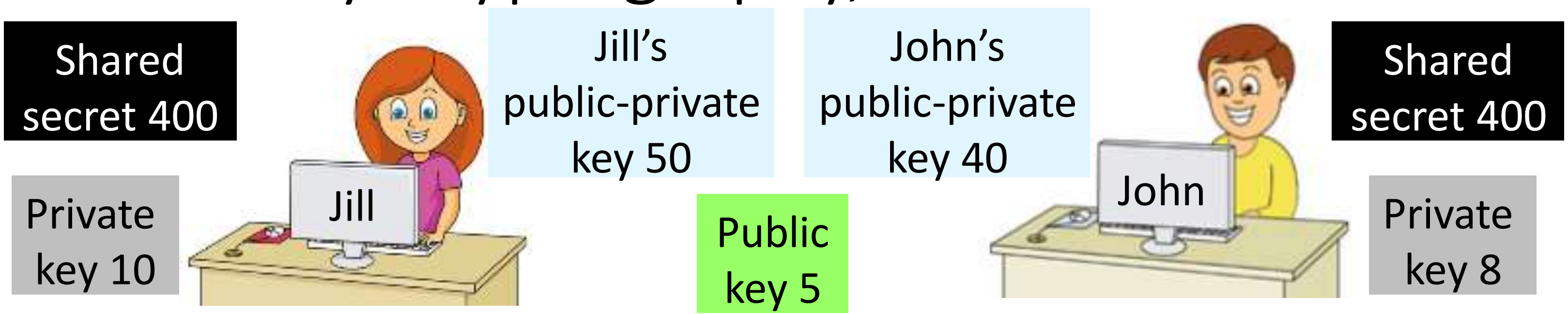
- We <u>cannot</u> discover John's private key 8 by dividing his public-private key 40 by the public key 5.

Public Key Cryptography, *cont'd*



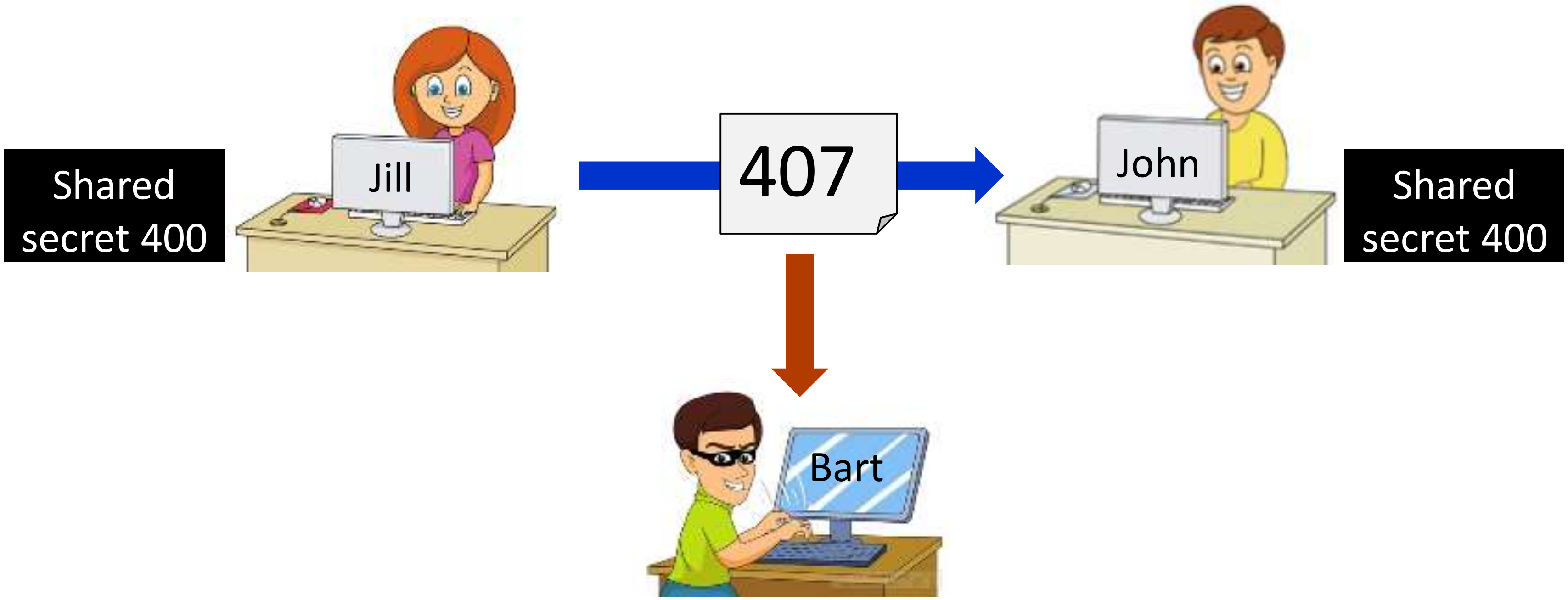
- What is the goal of all this?
  - To create a shared secret between Jill and John.
- Jill multiplies John's public-private key by her private key: 40x10=400
- John multiplies Jill's public-private key by his private key: 50x8=400

Public Key Cryptography, *cont'd*



- Now Jill and John have a shared secret 400.
- Jill can encrypt the confidential data 7 by <u>adding</u> the shared secret 400.
- John can decrypt the confidential data 7 by <u>subtracting</u> the shared secret 400.

Public Key Cryptography, *cont'd*



- Bart can't decrypt the 407 because he doesn't know the shared secret 400.

San José State University

Public Key Cryptography, *cont'd*

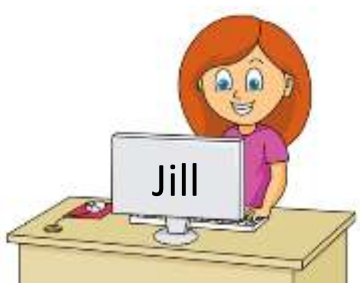
- Public key encryption works
  with <u>multiple recipients</u>.

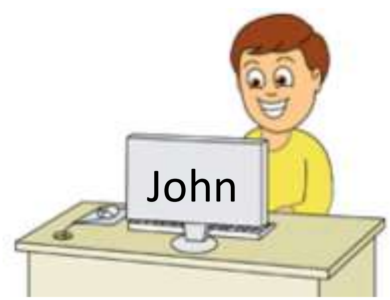
- Jill needs to send confidential data
  to both John and his
  twin brother Mark.

- Each picks a private key.

Private key 2

Private key 10

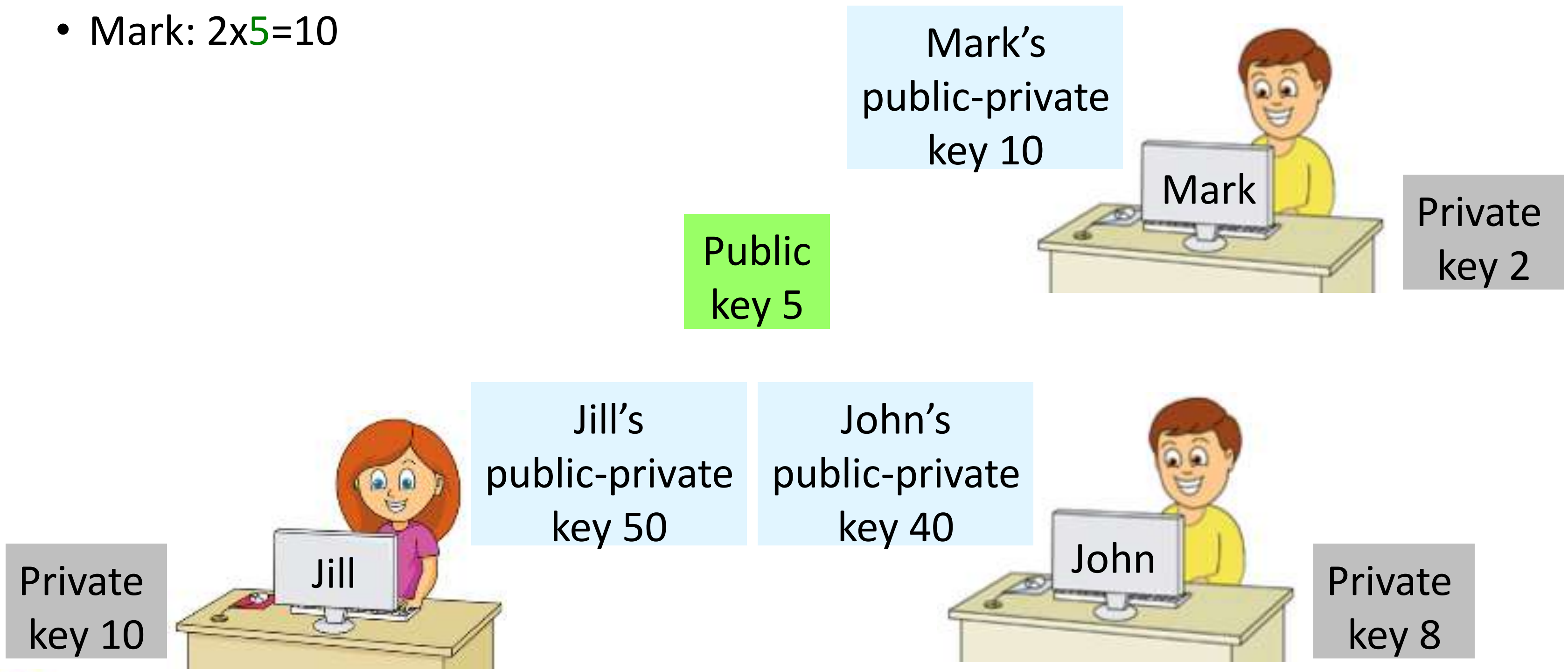




Private key 8

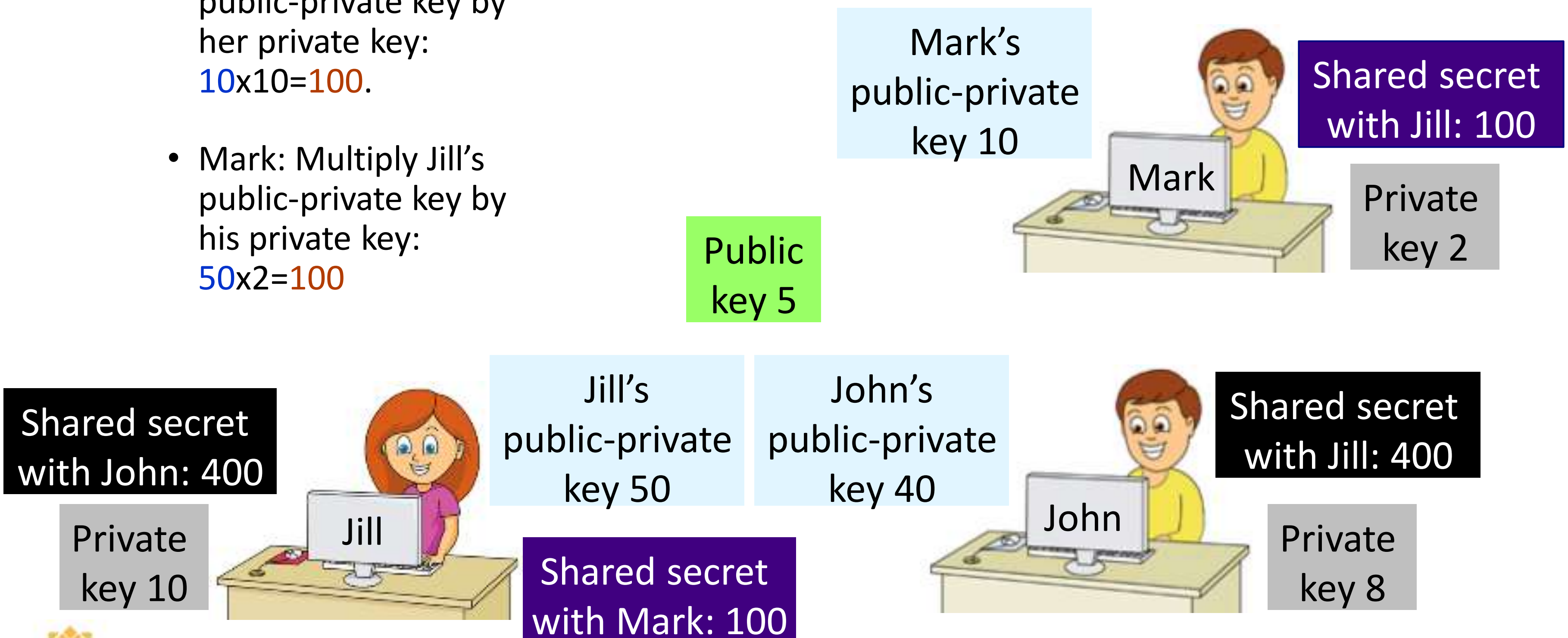
Public Key Cryptography, *cont'd*

- Jill announces the public key 5, and everyone generates his or her public-private key.
  - Jill: 10x5=50
  - John: 8x5=40
  - Mark: 2x5=10



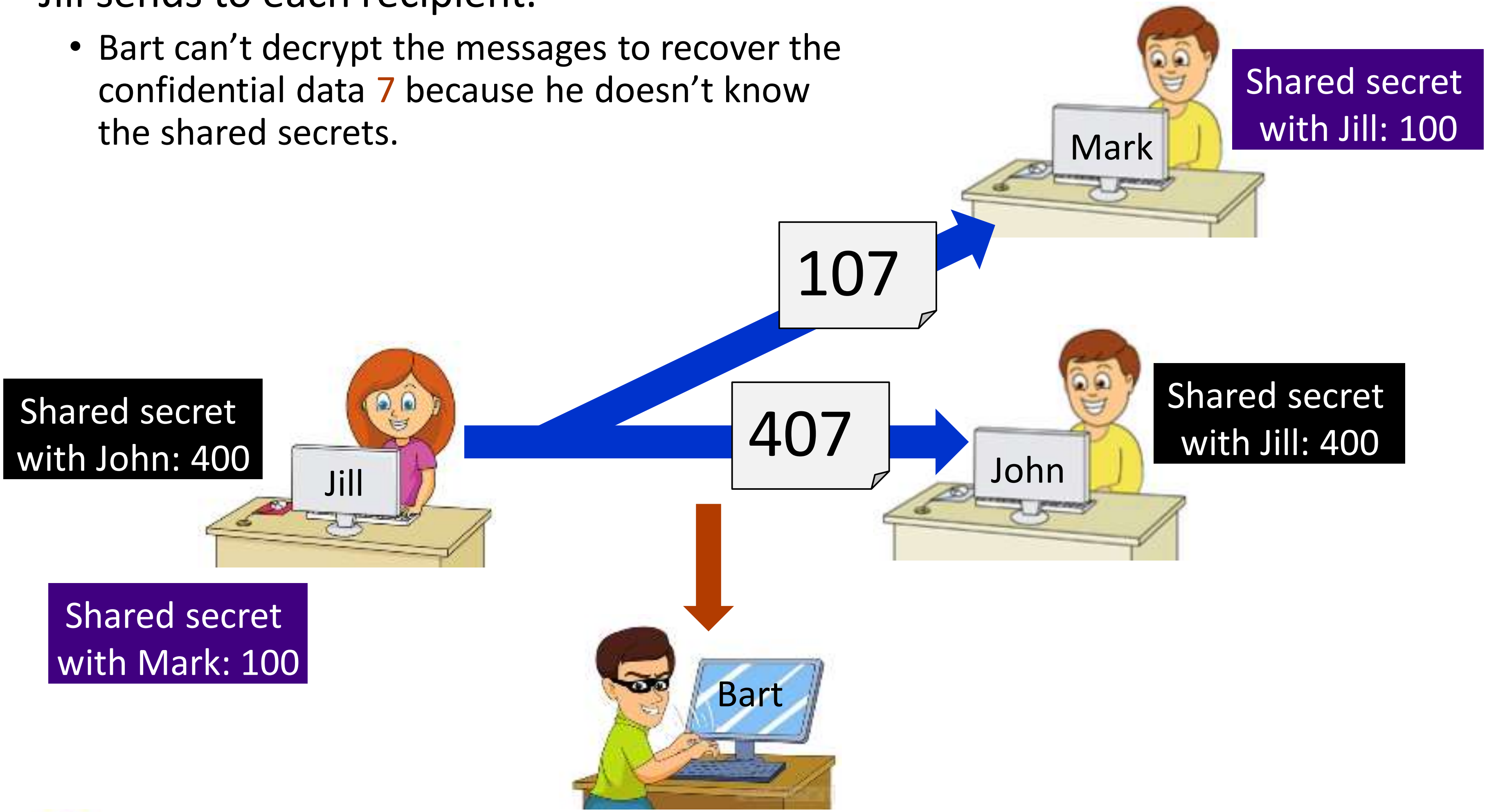
Public Key Cryptography, *cont'd*

- Jill will have a shared secret with each recipient.
  - Jill and John will share 400 between them, as before.
  - Jill and Mark will have a different shared secret.
    - Jill: Multiply Mark's public-private key by her private key: 10x10=100.
    - Mark: Multiply Jill's public-private key by his private key: 50x2=100



Public Key Cryptography, *cont'd*

- Jill sends to each recipient.
  - Bart can't decrypt the messages to recover the confidential data 7 because he doesn't know the shared secrets.



Cryptography in the Real World

- Of course, in the real world, we can't use simple operations like multiplication and addition to generate keys and to encrypt data.
  - Multiplication and addition are not one-way operations.

- Real-world encryption uses very large prime numbers and modulo arithmetic.
  - Not even today's most powerful supercomputer can undo such operations.
  - Worry: Can quantum computers in the future?

San José State University

When is Cryptography Used?

- Public key cryptography is a key exchange protocol first published by Whitfield Diffie and Martin Hellman in 1976.
  - It was actually invented earlier in 1970 by the British government, but it was classified.

- Whenever you visit a secure website, you are using the Diffie-Hellman protocol or a variant.
  - A secure website has a URL that starts with **`https:`** instead of **`http:`**

Computer Security as a Career

- Cybersecurity is a <u>hot</u> field.
  - Computers are used everywhere.
  - Big data.
  - Privacy issues.

San José State
UNIVERSITY