COMPUTER SECURITY

Key Exchange and Asymmetric Cryptography

Topics for today

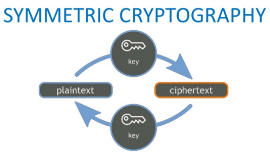
- Cryptography:
  - Problems encryption is designed to solve
  - Encryption tools categories, strengths, weaknesses
    - applications of each
  - Certificates and certificate authorities

KEY EXCHANGE

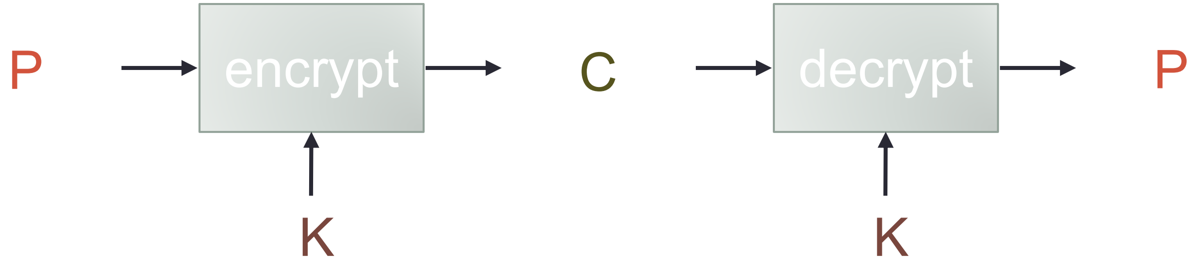
Cryptography - review

- Encryption provides secrecy
  - Confidentiality
- Symmetric cryptography requires a secret key
  - Need to be shared ahead of communication
    - Stored securely
  - Keys management a critical part of cryptography
    - One of the weakest points
      - Need to be refreshed, changed upon demand, destroyed

Reminder - Symmetric Cryptosystem



- Scenario
  - Alice wants to send a message (plaintext P) to Bob.
  - The communication channel is insecure and can be eavesdropped
  - If Alice and Bob have previously agreed on a symmetric encryption scheme and a secret key K
    - => the message can be sent encrypted (ciphertext C)



The Secret Keys

- ***Symmetric*** algorithms use one key, which works for both encryption and decryption
  - Usually, the decryption algorithm is closely related to the encryption one
    - running the encryption in reverse
- Both parties share a secret key
  - they can both encrypt sent information as well as decrypt information from the other

Key Exchange

- As long as the key remains secret, the system also provides authenticity
  - Authenticity is ensured because only legitimate sender can produce a message that will decrypt properly
    - with the shared key

Key Exchange

- How do two users obtain their shared secret key?
  - Need a sharing mechanism
- Once A and B share a key, only they can use that key for their encrypted communications
  - If A wants to communicate with another user C, A and C need a different shared secret key

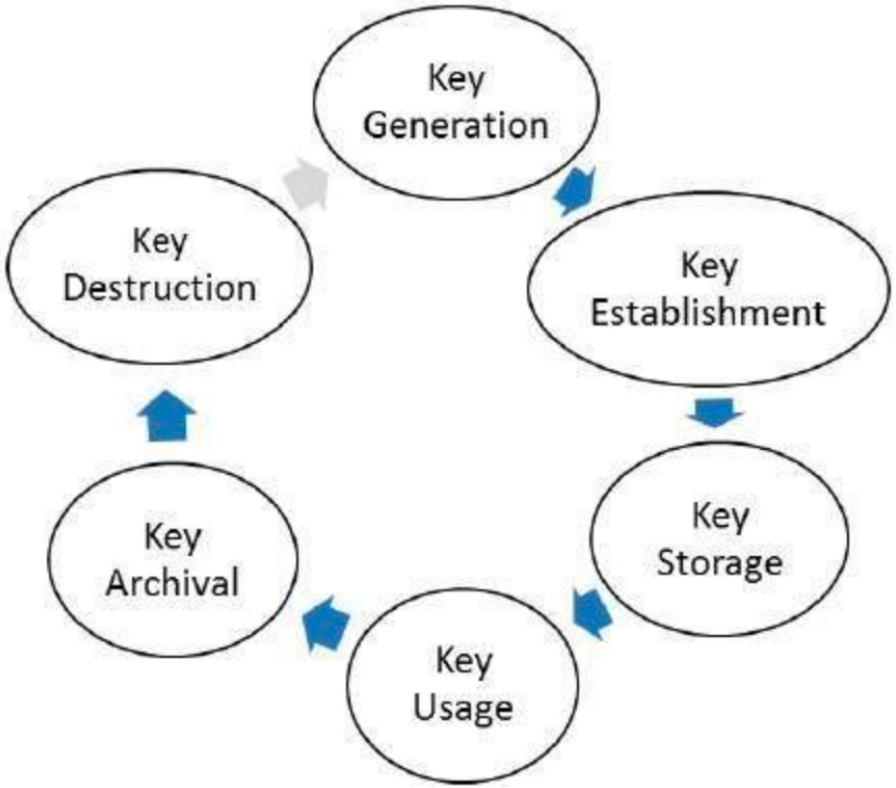
The Secret Keys

- If we have *n* users, how many keys do we need?
  - *n* users who want to communicate in pairs need a shared key for every pair of users
    - Total of $\binom{n}{2} = \frac{n \cdot (n-1)}{2}$ keys
  - This is $O(n^2)$, the number of keys grows at a rate proportional to the square of the number of users

Key Exchange

- Keys have a life-cycle
  - Cannot be reused forever, need to be refreshed
- Managing keys is the major difficulty in using symmetric encryption

Key Management



https://www.tutorialspoint.com/cryptography/public_key_infrastructure.htm

The Secret Keys

- Symmetric encryption systems require a means of key distribution
- How do we solve this?

Key Exchange with TTP

- One possibility:
  - Using a Third Trusted Party (TTP)
    - Classical cryptography
- In this case, each party will have one shared secret key with the third trusted party
  - $K_a, K_b, K_c, etc.$

Key Exchange with TTP

- What A and B want to exchange a secret key:
  - TTP picks a new secret key $K_{a,b}$
  - TTP encrypts $K_{a,b}$ with K_a and sends it to A
  - TTP encrypts $K_{a,b}$ with K_b and sends it to B
  - A and B each decrypts their respective keys
    - Using their pre-shared secret key

Key Exchange with TTP

- C = Third trusted party
- C shares a secret K_a and K_b with Alice and Bob
- Key exchange protocol:
- 1. $A \rightarrow C: \{Request\ K_{a,b}\}_{K_a}$
- 2. $C \rightarrow A: \{K_{a,b}\}_{k_a}, \{K_{a,b}\}_{k_b}$
- 3. $A \rightarrow B: \{K_{a,b}\}_{k_b}$

Key Exchange with TTP

- Disadvantage: TTP always has to be available online to perform key exchange
- Is there another method?
  - Yes, using ***asymmetric encryption***

Key Exchange

- ***Asymmetric or public key*** systems typically have precisely matched pairs of keys.
- The keys are produced together
  - One may be derived mathematically from the other
  - Process computes both keys as a set

Key Exchange

- ***Asymmetric systems*** good for key management
  - public key may be emailed or post it in a public directory
- Only the corresponding private key can decrypt what has been encrypted with the public key

Key Exchange with Asymmetric Cryptography

- Alice chooses a pair of public and private keys K_{pb} and k_{pr}
- Alice distributes K_{pb} public key to all parties
- Bob chooses a secret key $K_{b,a}$ and encrypts it with K_{pb}
  - Sends $E(K_{b,a}, K_{pb})$ to Alice
- Alice uses her secret key K_{pr} to decrypt $K_{b,a}$
- Alice and Bob now share a secret key $K_{b,a}$

PUBLIC KEY (ASYMMETRIC) ENCRYPTION

Motivation

- “The basics of asymmetric cryptography are fundamental concepts that any member of society who wants to understand how the world works, or could work, needs to understand. They are as fundamental as the basics of supply and demand and monetary inflation”

Phil Libin, Evernote

Public Key Encryption

- A cryptographic system that uses pairs of keys
  - public keys which may be disseminated widely
    - Any person can encrypt the message
  - private keys which are known only to the owner
    - Message can only be decrypted with this key
- Analogous to a self-closing door
  - Need key to open it
  - Closing it shuts it automatically

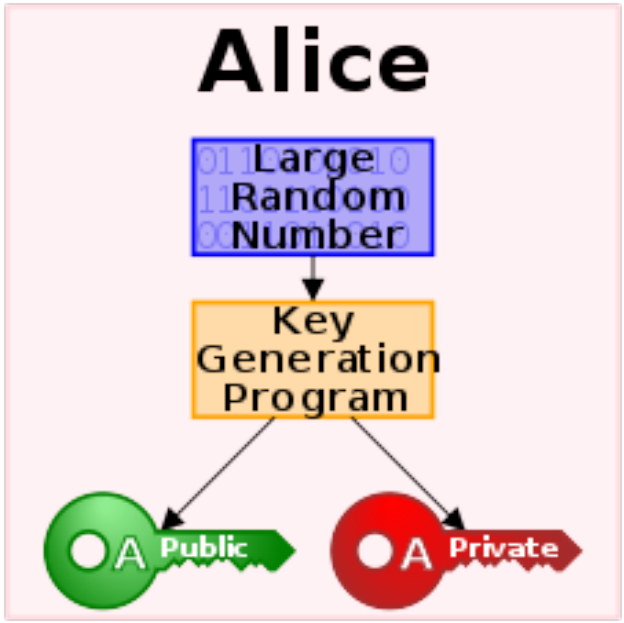
Public Key (Asymmetric) Cryptography

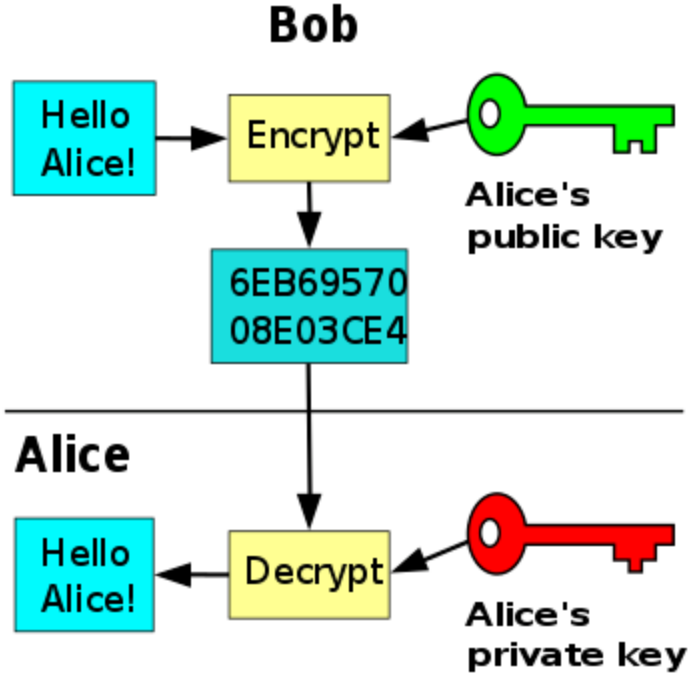
- Instead of two users sharing one secret key, each user has two keys: one public and one private
- Messages encrypted using the public key can only be decrypted using the private key
  - $P = D(K_{priv}, E(K_{pub}, P))$

Public Key Encryption

- Why does it work?
- It is not feasible to compute the private key
  - From knowledge of its paired public key
- Therefore, only the private key is kept private
  - The public key can be openly distributed without compromising security
- Public key cryptography relies on math problems that have no efficient solution

Public Key Encryption



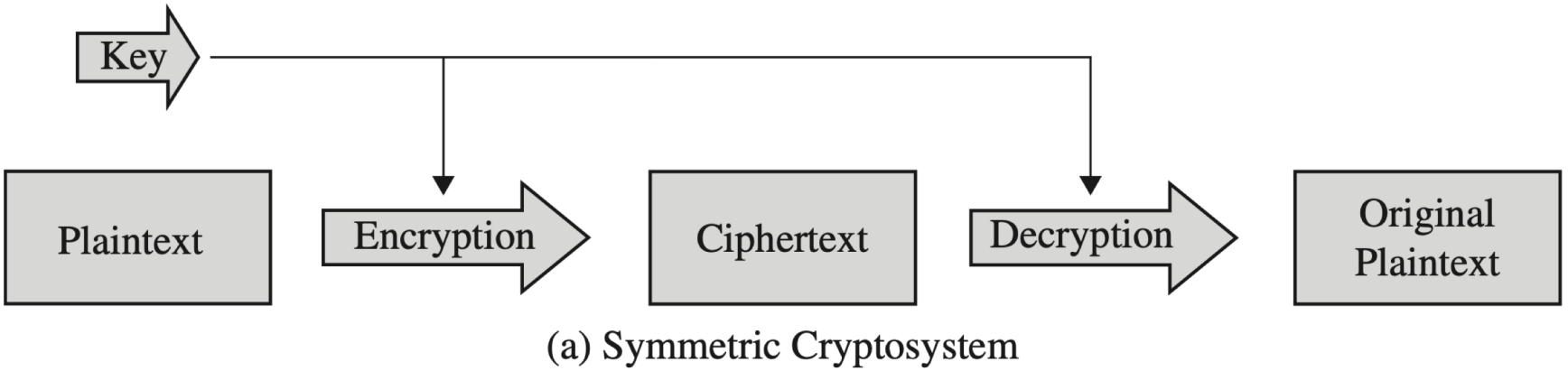


https://en.wikipedia.org/wiki/Public-key_cryptography

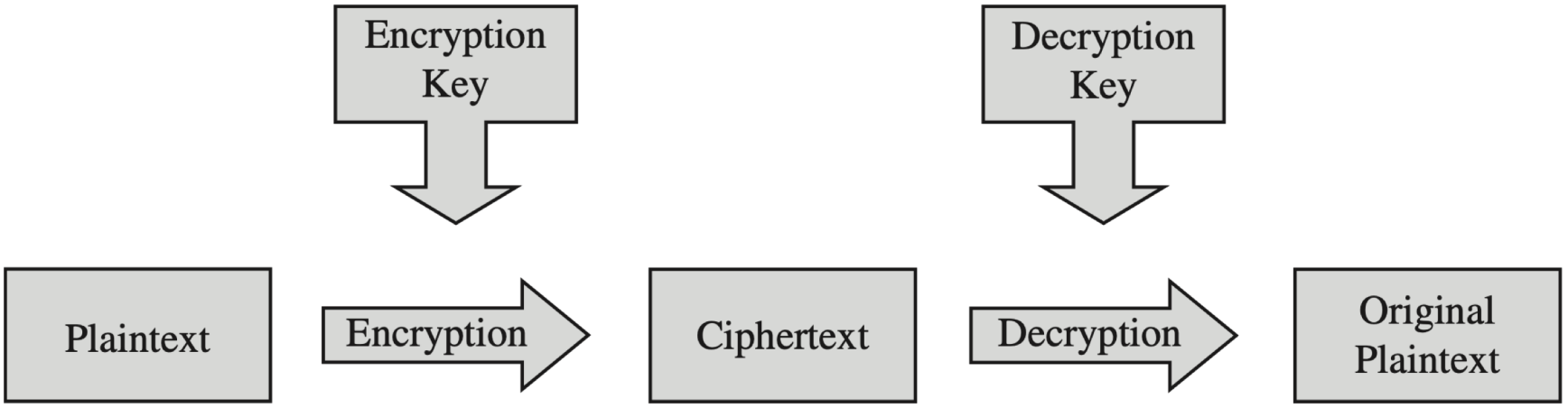
Public Key Encryption

- Often used to secure communication
  - Over the internet, open networks
  - Typically used for key exchange

Symmetric vs. Asymmetric



(a) Symmetric Cryptosystem

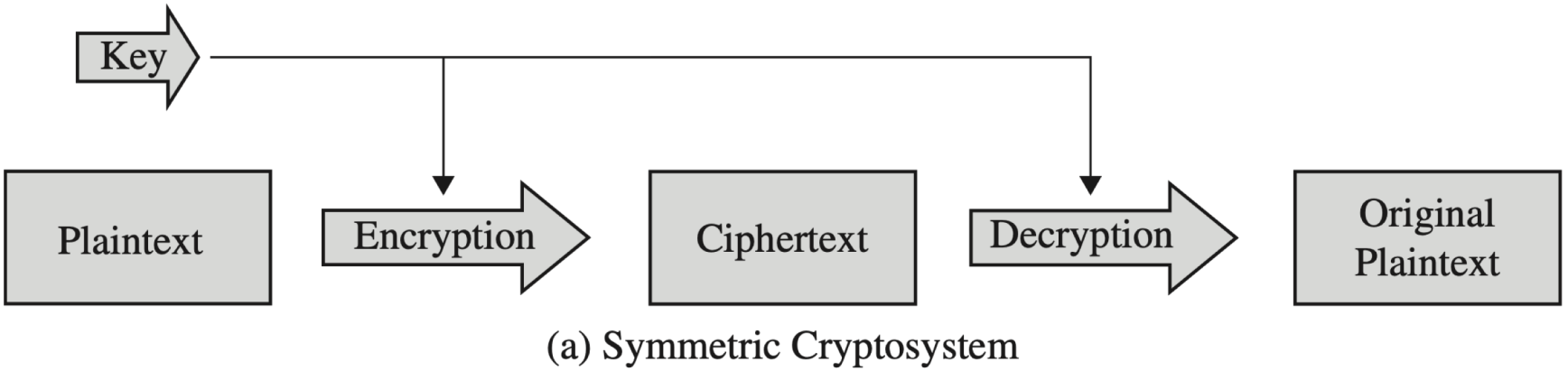


(b) Asymmetric Cryptosystem

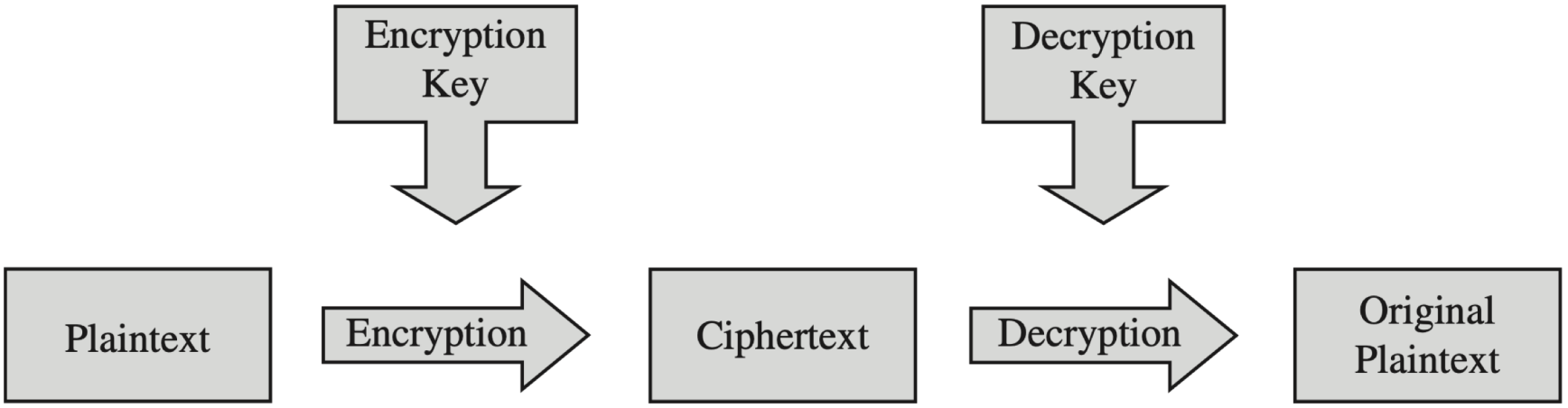
Secret Key vs. Public Key Encryption

	Secret Key (Symmetric)	Public Key (Asymmetric)
Number of keys	1	2
Key size (bits)	56–112 (DES), 128–256 (AES)	Depending on cryptosystem, from 256 to 10000 and more
Protection of key	Must be kept secret	One key must be kept secret; the other can be freely exposed
Best uses	Cryptographic workhorse. Secrecy and integrity of data, from single characters to blocks of data, messages and files	Key exchange, authentication, signing
Key distribution	Must be out-of-band	Public key can be used to distribute other keys
Speed	Fast	Slow, typically by a factor of up to 10,000 times slower than symmetric algorithms

Symmetric vs. Asymmetric



(a) Symmetric Cryptosystem



(b) Asymmetric Cryptosystem

Symmetric vs. Asymmetric

- The critical difference between symmetric and asymmetric cryptography:
  - Symmetric uses a single key for both encryption and decryption
    - whereas asymmetric uses complementary keys

The RSA Cryptosystem (Rivest-Shamir-Adelman, 1978)

- The most widely used public-key cryptosystem
  - But becoming less popular recently
- Security relies on the hardness of the integer factorization problem
  - Given a large integer, it is costly to recover its factors
  - Makes is hard to recover secret key from public key

The RSA Cryptosystem (Rivest-Shamir-Adelman, 1978)

- Keys must be very long
  - At least 1024-bit long, usually 2048-8192
  - Because the factorization problem is not hard enough when the integers are smaller
  - Will cover it later in the course

RSA Runtime

- Keys are long: 1024-8192 bits
- Encryption is done by modular exponentiation
  - A complicated mathematical operation,

$$x, e, N \mapsto x^e mod\ N$$

  - Significantly slower that substitution, transposition that are used in symmetric encryption

- RSA runtime significantly longer than DES, AES, etc.

Diffie-Hellman

- One of the first key exchange protocols
  - Designed solely for key exchange

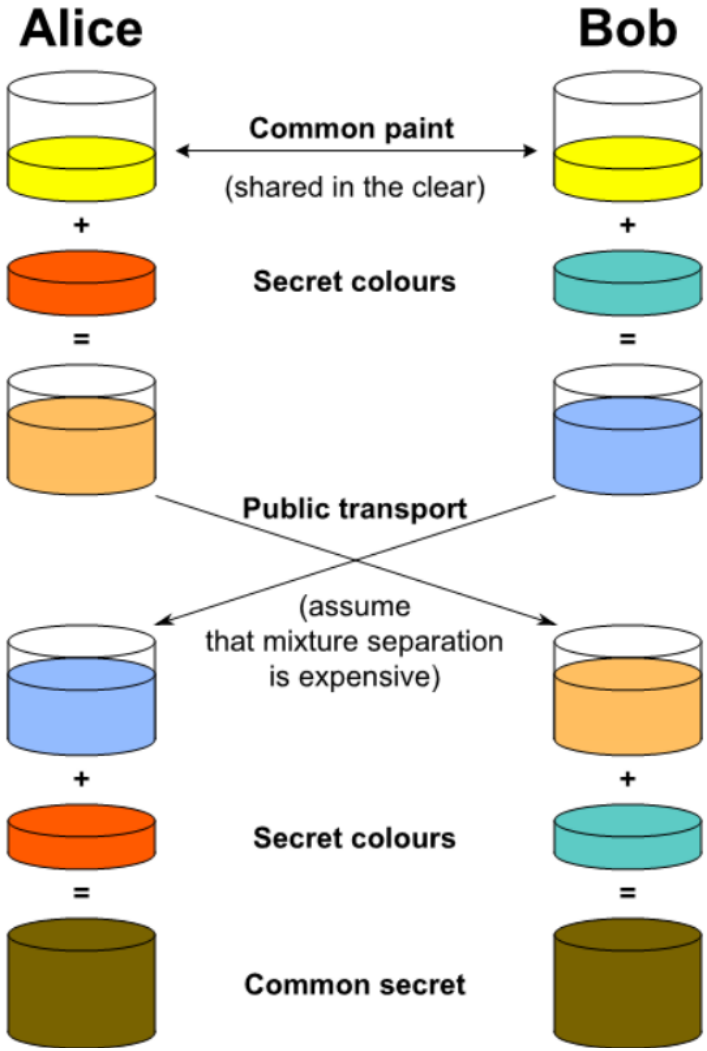
Diffie-Hellman Protocol

- Two users agree on one shared large integer number
  - Does not need to be secret
- Each user chooses another large random number
  - These need to be secret
- Each user combines the shared number and their secret number
  - Send the result to each other

Diffie-Hellman Protocol

- Each user combines their secret number with the result they received from the other person
  - Result will be the same on both sides

Diffie-Hellman Protocol



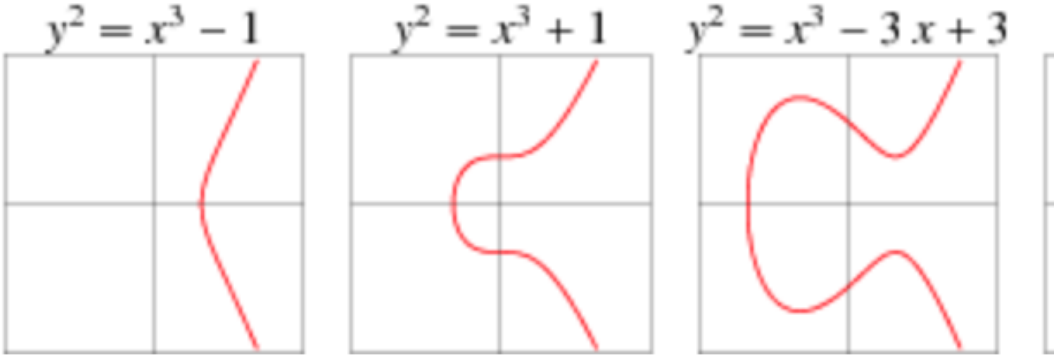
https://www.codeproject.com/Articles/1219531/Implement-Diffie-Hellman-in-Csharp

Elliptic Curve Cryptography (ECC)

- A public key encryption system
- Uses algebraic structure of finite fields to generate secure keys
- Able to achieve a similar level of security to traditional public key algorithms
  - With smaller key size

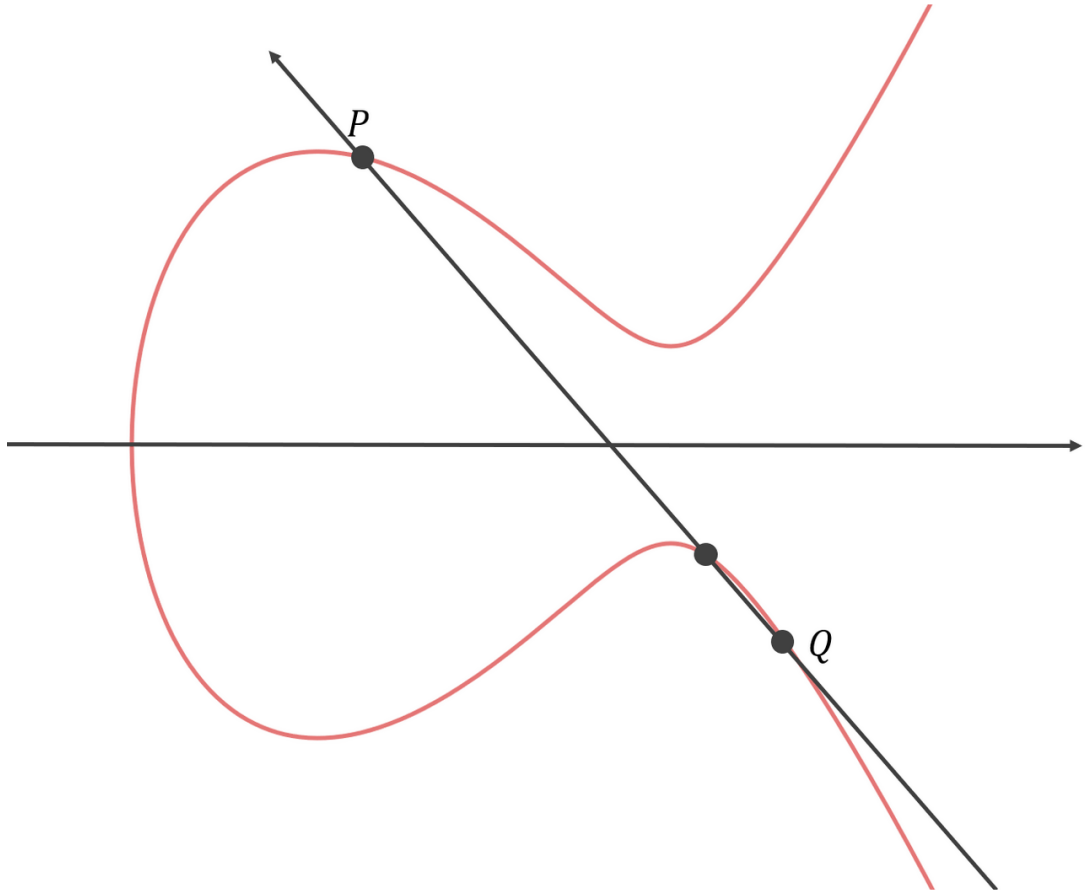
Elliptic Curve Cryptography (ECC)

- Elliptic curve defined by: $y^2 = x^3 + ax + b$
- Non-vertical lines will intersect curve at most in 3 places



https://hackernoon.com/what-is-the-math-behind-elliptic-curve-cryptography-f61b25253da3

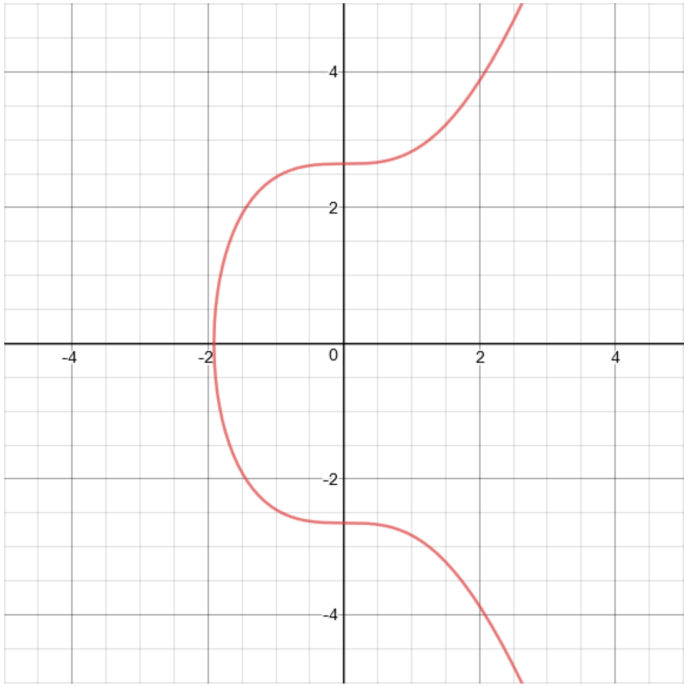
Elliptic Curve Cryptography (ECC)



https://hackernoon.com/what-is-the-math-behind-elliptic-curve-cryptography-f61b25253da3

Example – ECC used by Bitoin

- $y^2 = x^3 + ax + b$



Elliptic-Curve Encryption

- ECDH is an elliptic curve variant of Diffie-Hellman
- Both NIST and the NSA recommend using ECC

Secret Key vs. Public Key Encryption

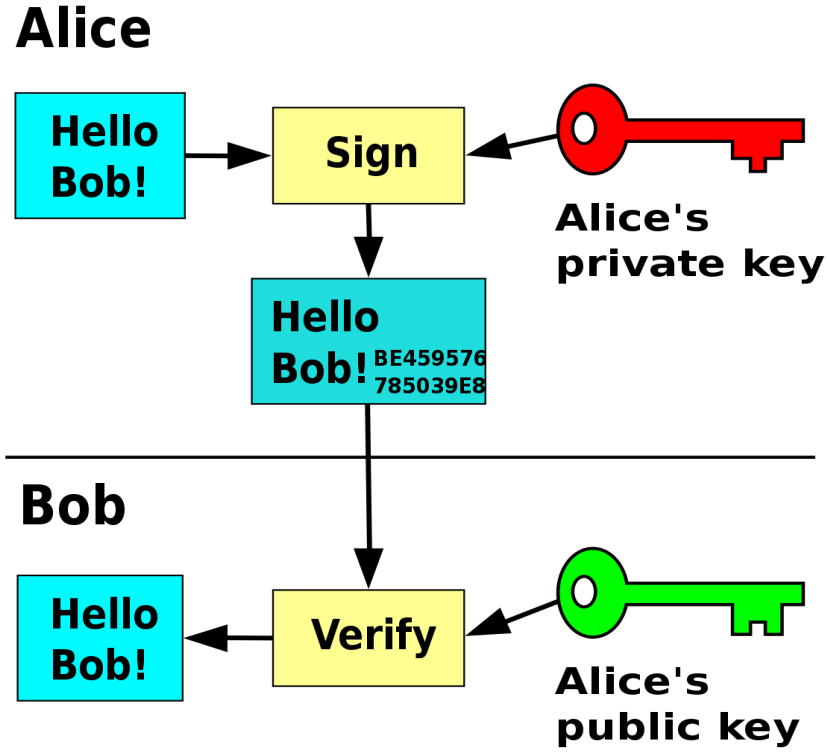
- Symmetric and asymmetric algorithms have complementary strengths and weaknesses
- Used for different purposes
  - in concert with each other

Secret Key vs. Public Key Encryption

	Secret Key (Symmetric)	Public Key (Asymmetric)
Number of keys	1	2
Key size (bits)	56–112 (DES), 128–256 (AES)	Depending on cryptosystem, from 256 to 10000 and more
Protection of key	Must be kept secret	One key must be kept secret; the other can be freely exposed
Best uses	Cryptographic workhorse. Secrecy and integrity of data, from single characters to blocks of data, messages and files	Key exchange, authentication, signing
Key distribution	Must be out-of-band	Public key can be used to distribute other keys
Speed	Fast	Slow, typically by a factor of up to 10,000 times slower than symmetric algorithms

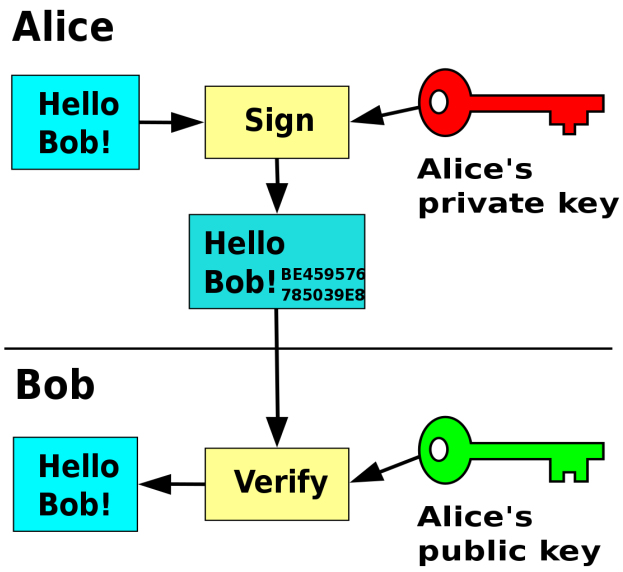
Digital Signatures

- Another use of public-key cryptosystems
- Use secret key to sign a message, public key to check validity of signature



Digital Signatures

- Another use of public-key cryptosystems
- Use secret key to sign a message, public key to check validity of signature
  - People sometime mistakenly refer to signature as “decryption”



Digital Signatures

- Three purposes:
  - ***Authentication***: message was created and sent by claimed sender.
  - ***Non-repudiation***: sender can not deny having sent the message later
  - ***Integrity***: ensures message not altered in transit

Digital Signatures

- Commonly used in:
  - Financial transactions
  - Software distribution
  - Other cases where forgery detection is needed
  - Popular in email applications

Digital Signatures

- Tool to demonstrate authenticity
  - similar to a paper signature
- A way by which a person or organization can affix a bit pattern to a file
  - implies confirmation,
  - pertains to that file only
  - cannot be forged

Digital Signatures

- A digital signature often uses asymmetric or public key cryptography
  - public key cryptographic protocols involve several sequences of messages and replies
    - Time consuming if both parties are not immediately available
  - In this situation, a technique that could authenticate a party even if it is inactive would be useful
    - Similar to a paper signature

Paper Signatures

- Traditional properties of a check signature:
  - A check is a *tangible object*
    - authorizing a financial transaction.
  - The check signature confirms authenticity
    - (presumably) only legitimate signer can produce that signature.
  - In the case of an alleged forgery, a third party can be called in to judge authenticity.

Paper Signatures

- Traditional properties of a check signature (cont.):
  - Once a check is cashed, it is canceled
    - it cannot be reused.
  - The paper check is not alterable.
    - most forms of alteration are easily detected.

Digital Signatures

- Properties required of digital signatures:
  - It must be unforgeable
    - If person S signs message M with signature Sig(S,M), no one else can produce the pair [M,Sig(S,M)].
  - It must be authentic
    - If a person R receives the pair [M, Sig(S,M)] purportedly from S, R can check that the signature is really from S.
    - Only S could have created this signature
      - the signature is firmly attached to M.

Digital Signatures

- Desired Properties:
  - It is not alterable
    - After being transmitted, M cannot be changed
      - By either S, R, or an interceptor.
  - It is not reusable
    - A previous message presented again will be instantly detected by R

Digital Signatures

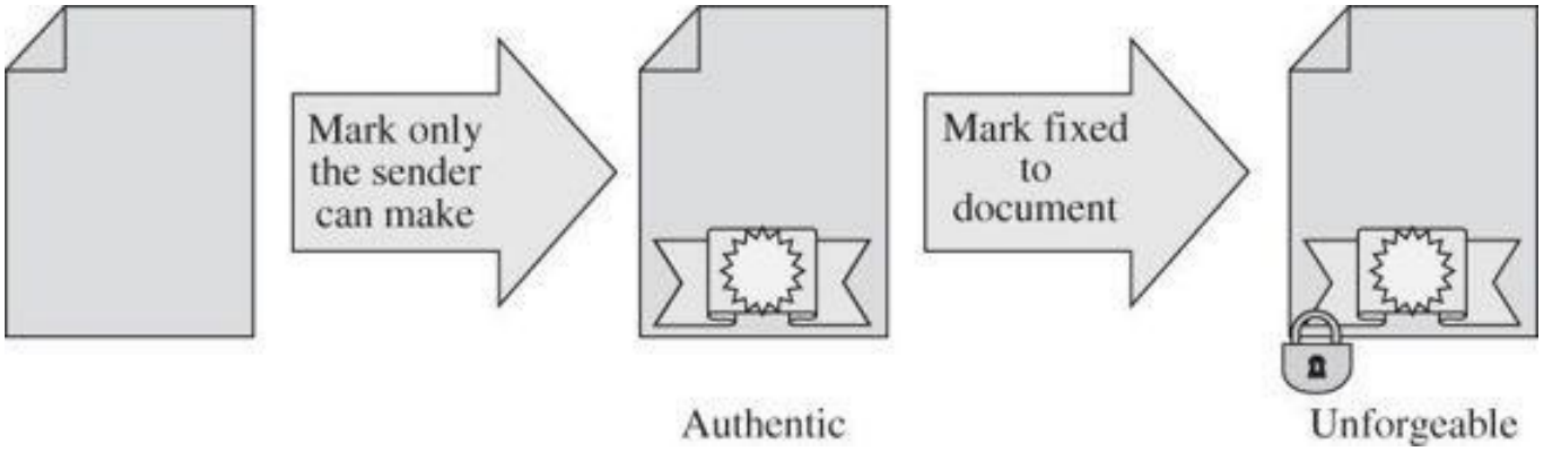
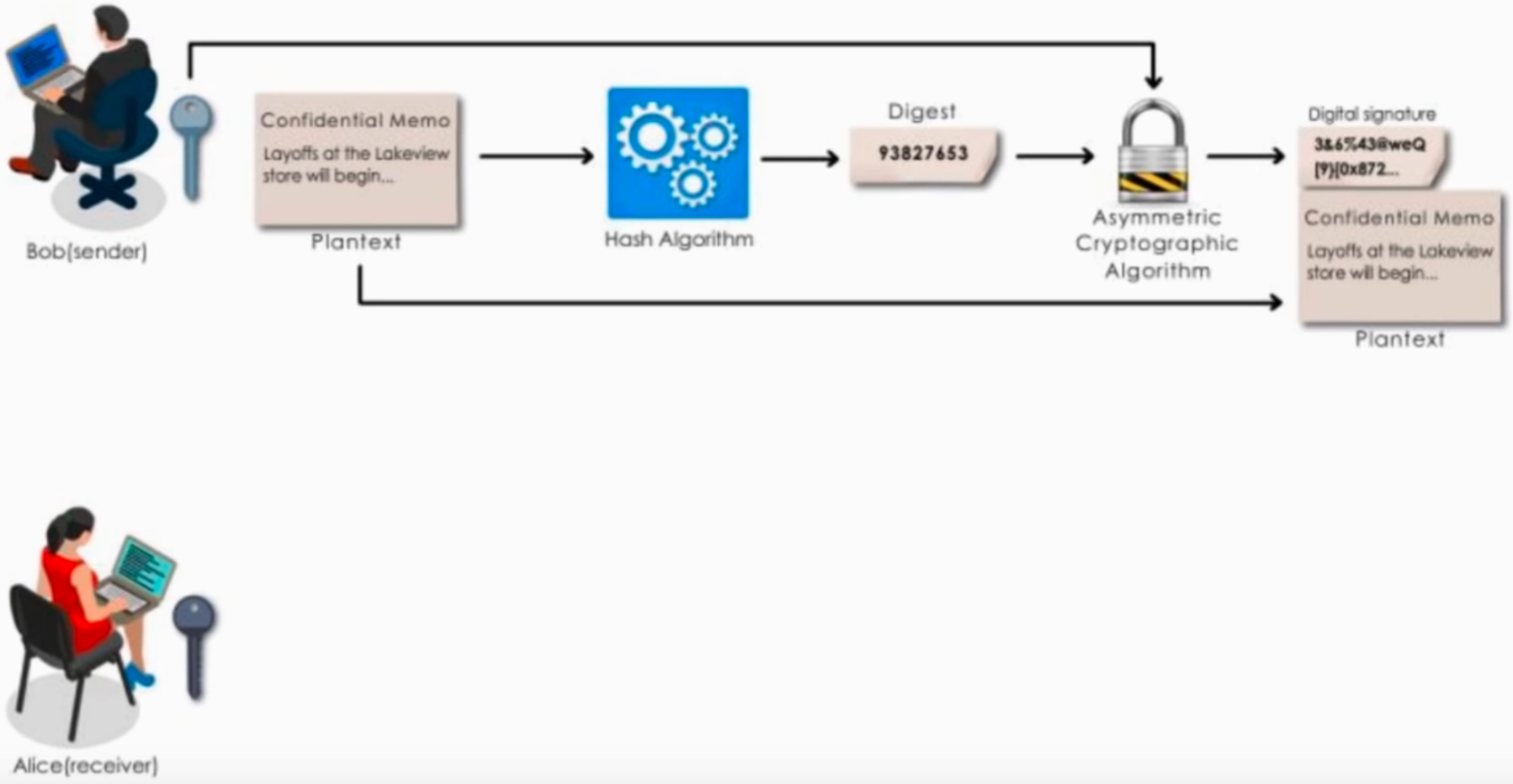


FIGURE 2-26 Digital Signature Requirements

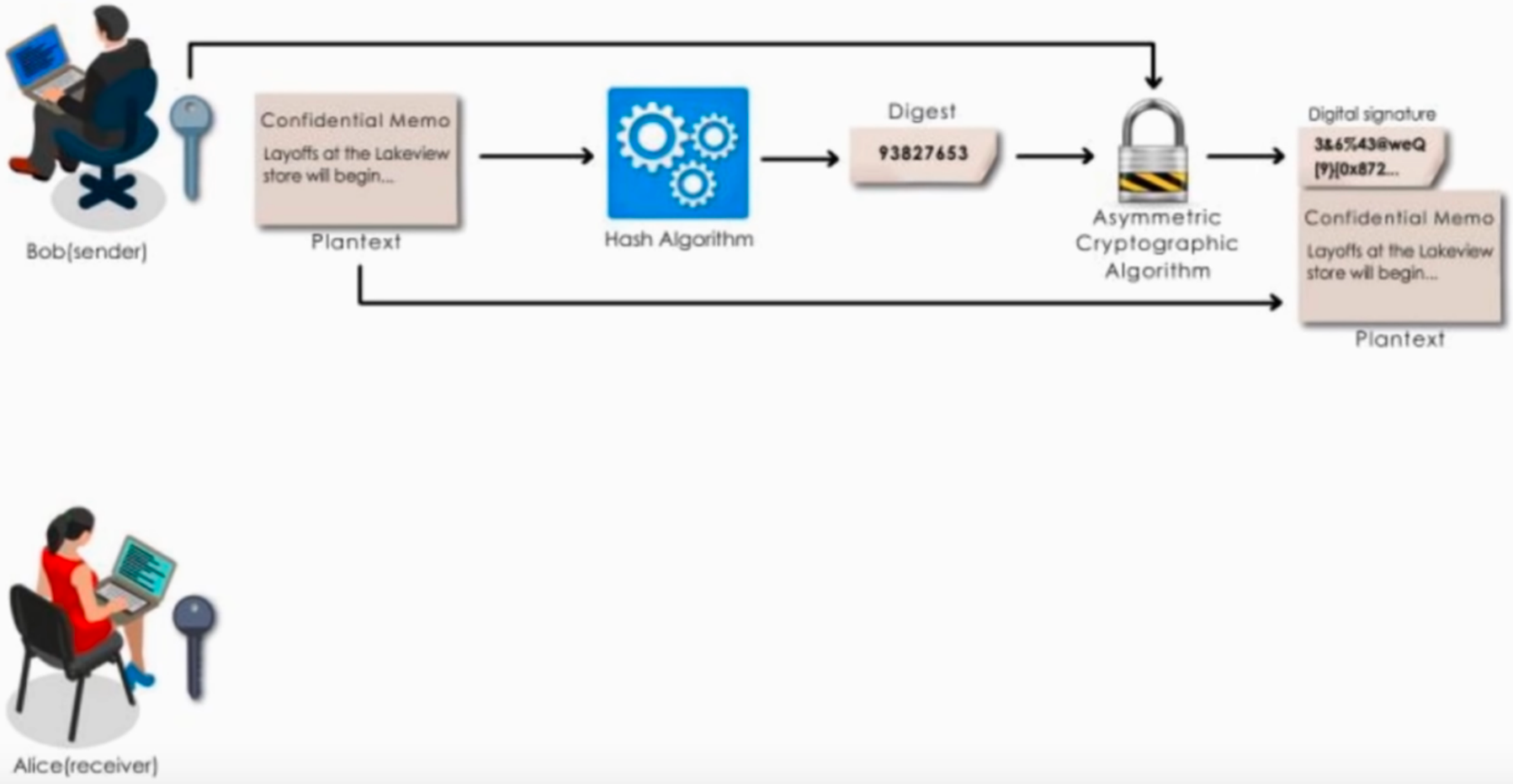
Digital Signature



Digital Signature

- Bob wants to sign a document
  - Bob generates a public and private key
    - Sends public key to Alice
  - Bob hashes the message to get a digest
  - Bob signs the digest using his private key
    - The digest is the digital signature for the memo
  - Bob sends the digital signature and memo to Alice

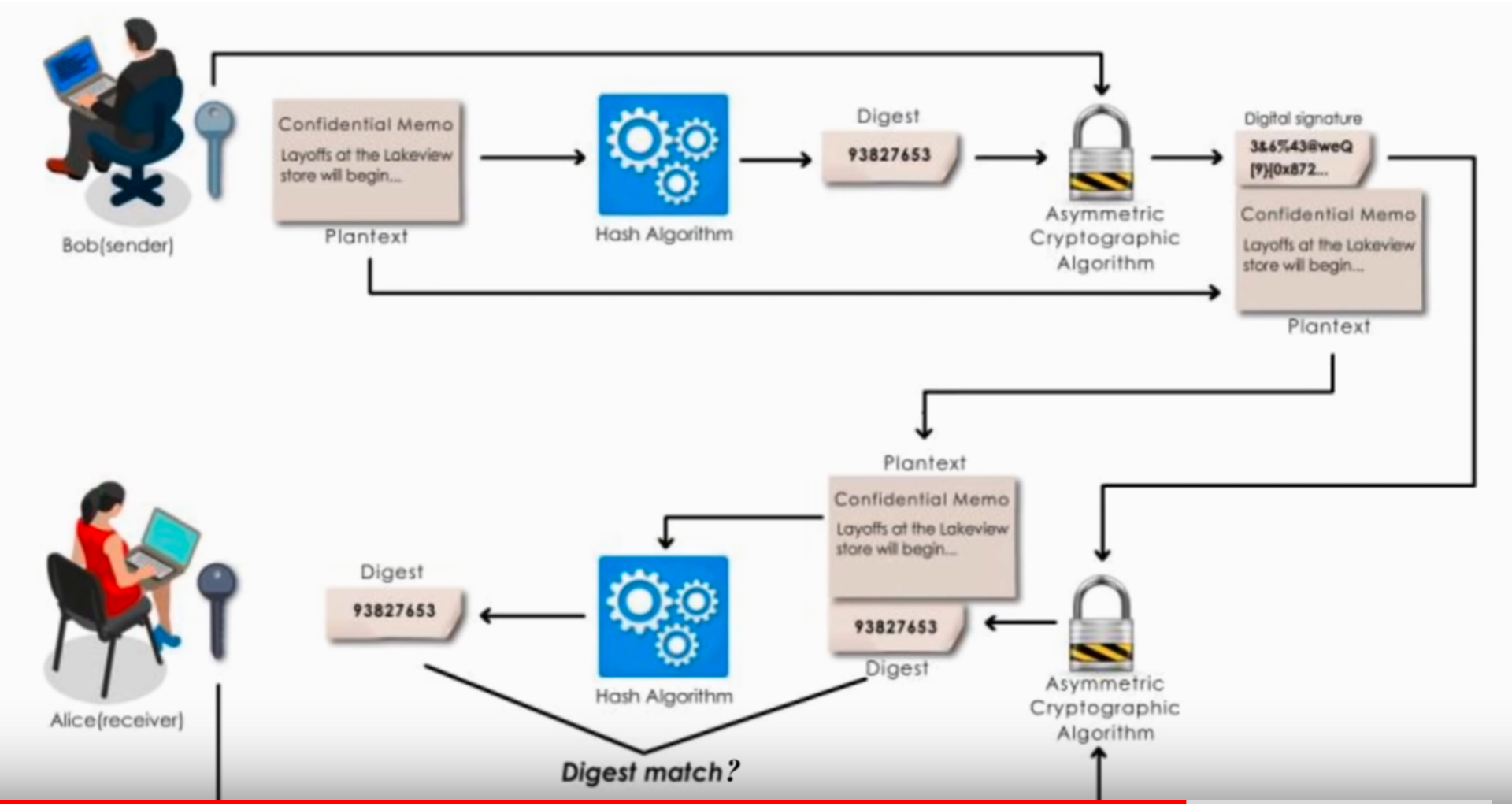
Digital Signature



Digital Signature

- Alice verifies the digital signature (utilizing Bob's public key)
  - If she can not verify it, she knows it did not come from Bob
- Alice hashes the plaintext and compares the digests
  - If they are equal, she knows message did not change

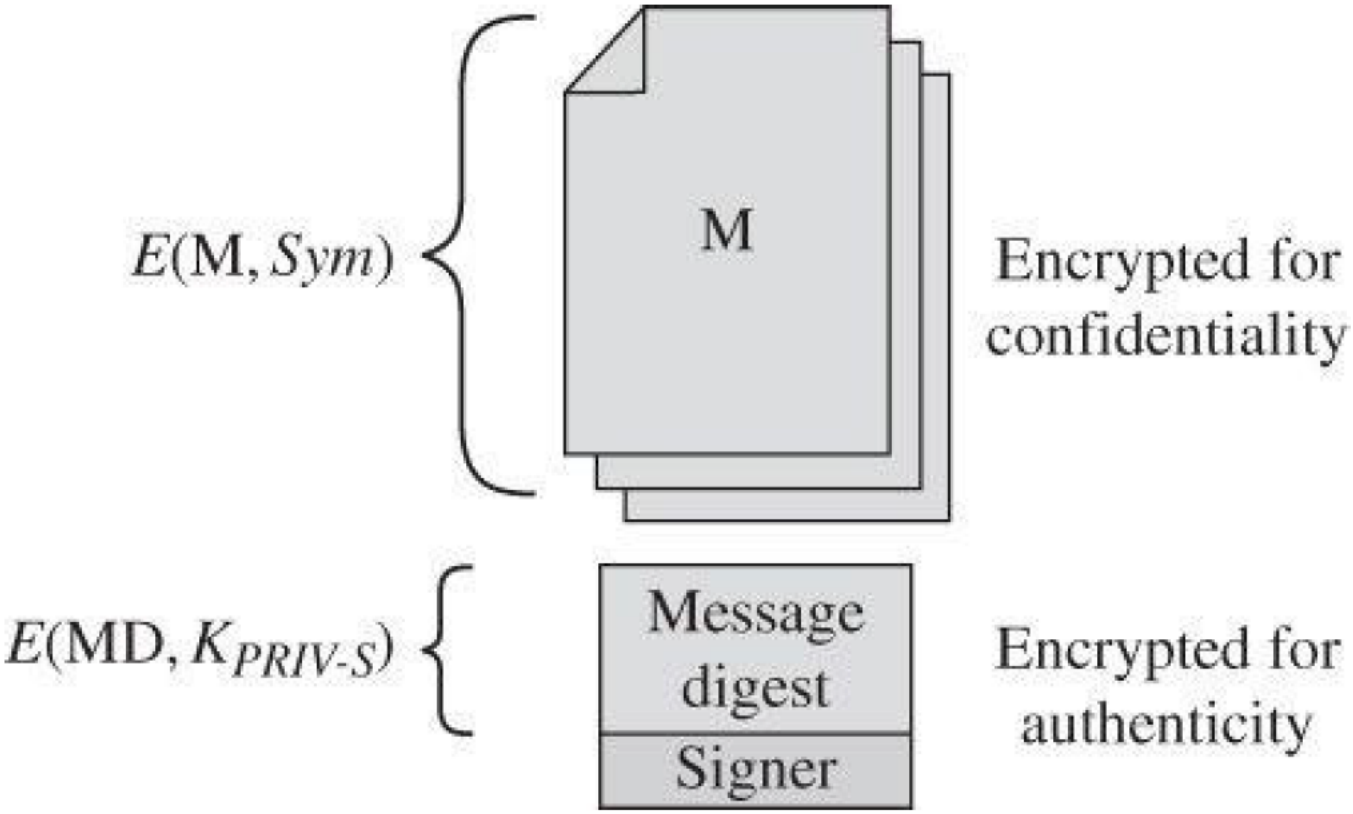
Digital Signature



Complete Digital Signature

- Confidentiality:
  - Digital signature does not provide confidentiality
    - In the previous example, message sent in plaintext
  - If confidentiality is required, signer can encrypt the file
    - Using symmetric key encryption and the user key

Complete Digital Signature



Digital Signatures

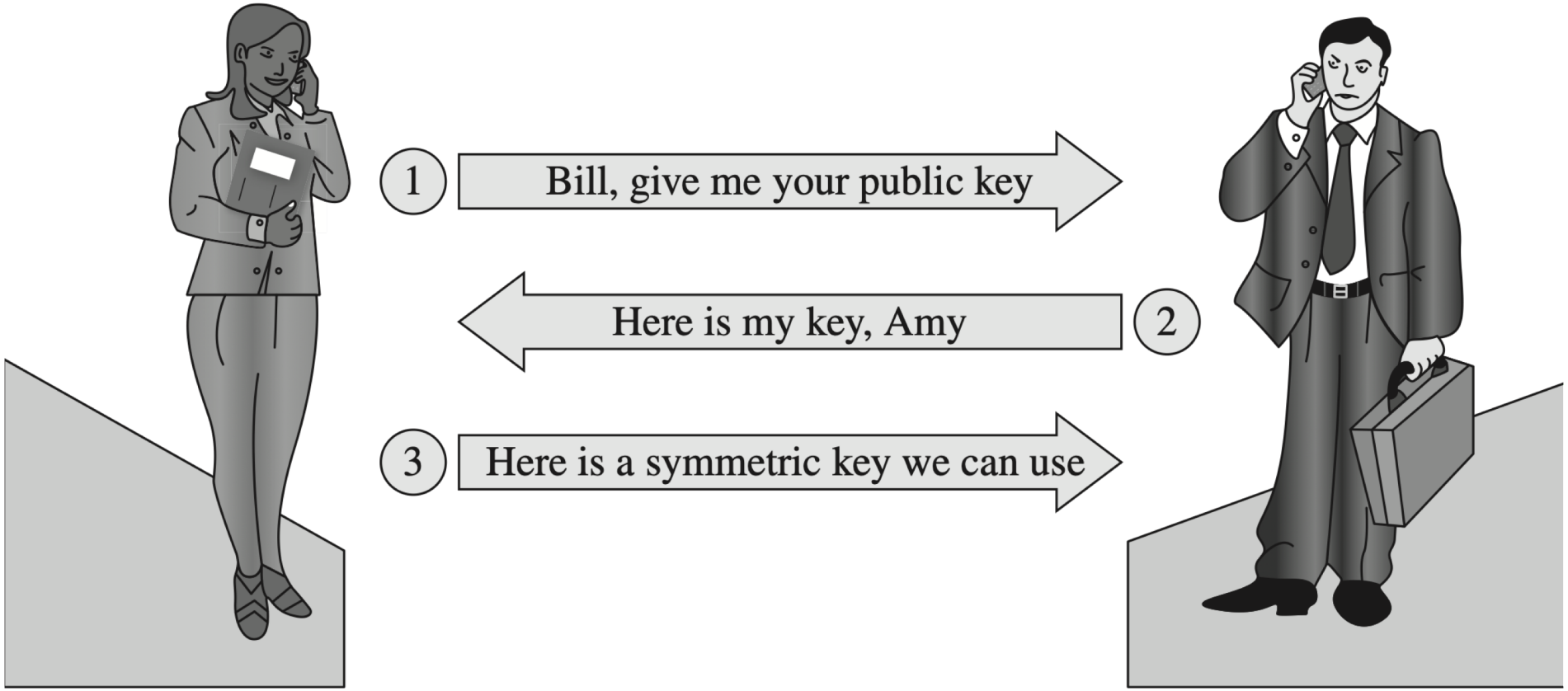
- A digital signature can indicate the authenticity of a file or a piece of code
- When installing a piece of signed code:
  - the operating system will inspect the certificate and file
  - notify you if the certificate and hash are not acceptable
- Digital signatures provide an effective tool
  - coupled with strong hash functions and symmetric encryption
  - ensure that a file is precisely what the originator stored for download

Digital Signature

- Digital Signature

BACK TO KEY EXCHANGE

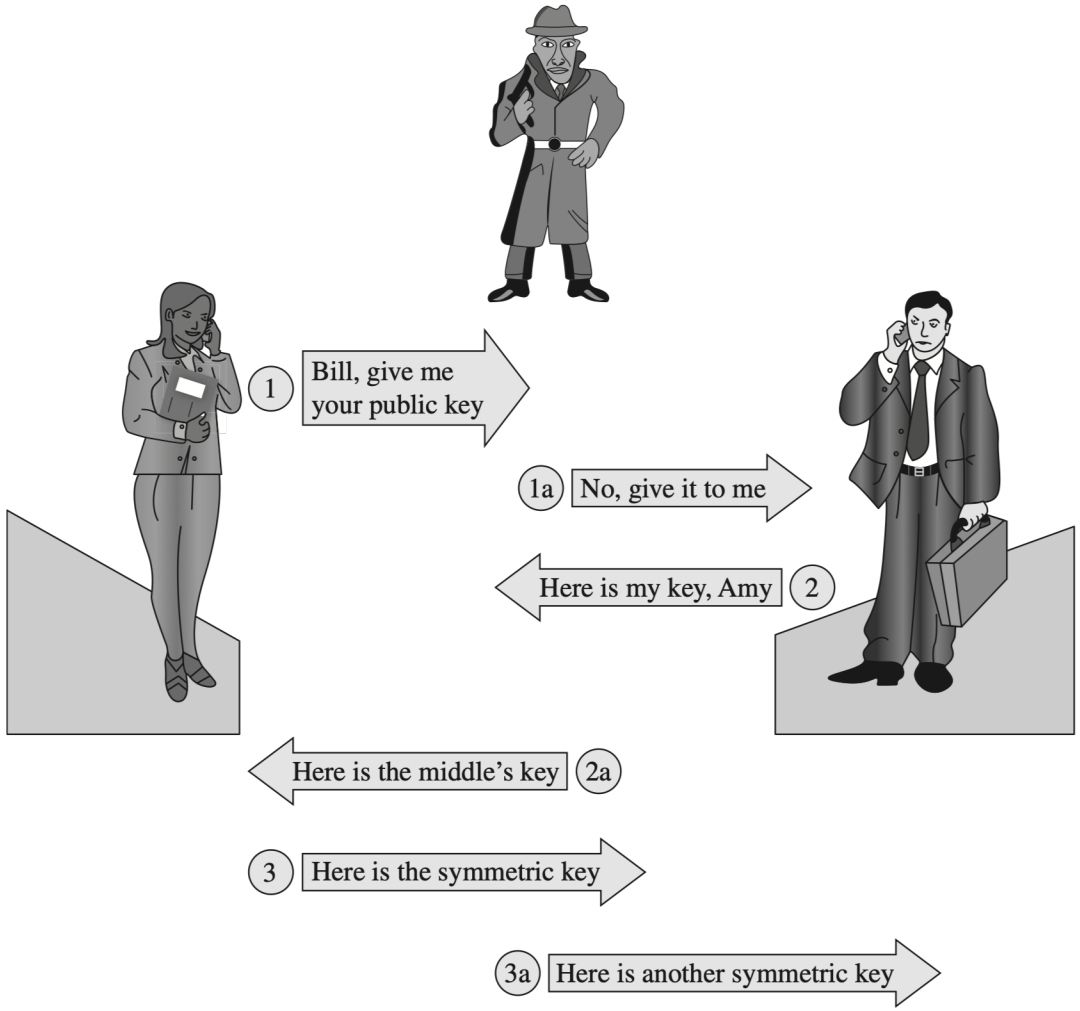
Public Key to Exchange Secret Keys



Public Key to Exchange Secret Keys

- What is our threat model?
  - To have a secure key exchange, users need to know something about each other
  - Typically, users will know each other's public key
  - Then, they can be used to sign the key exchange
- What if this is not the case?

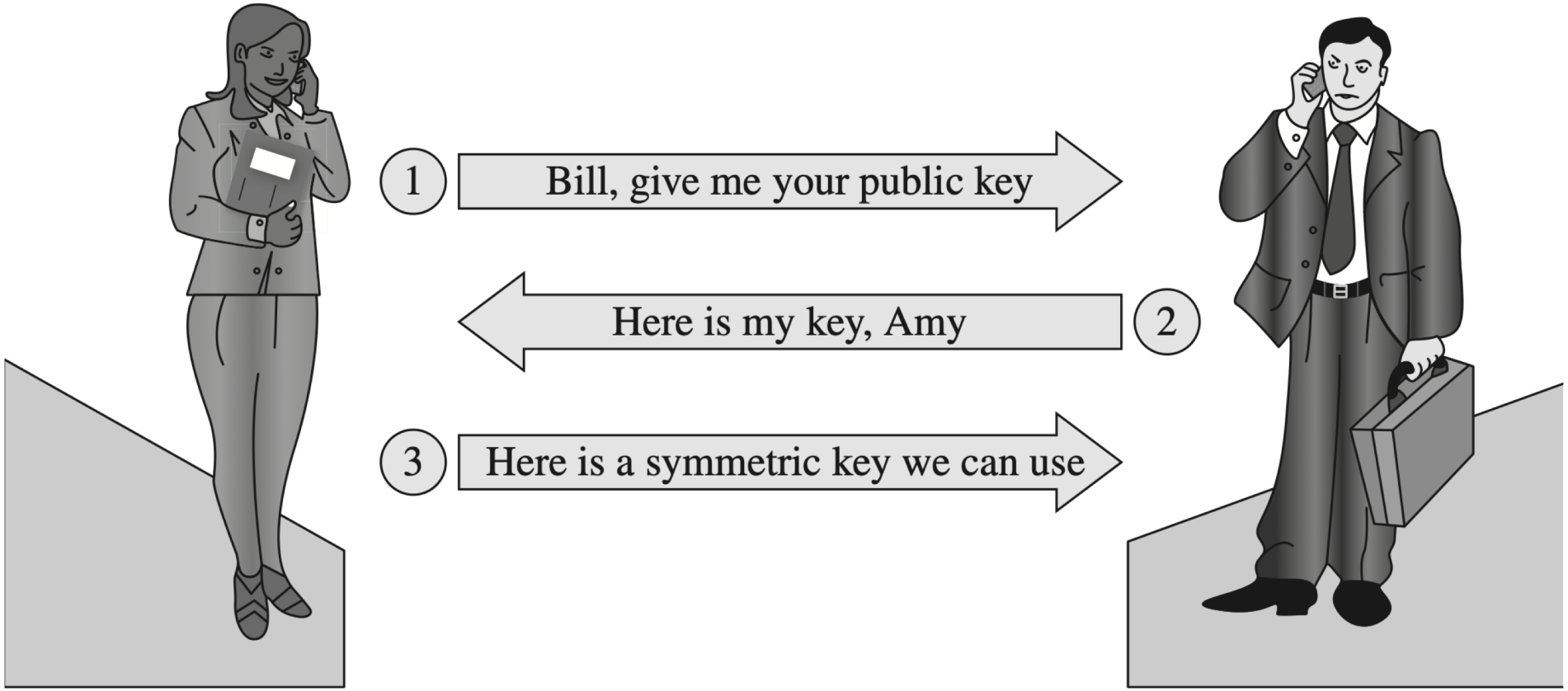
Key Exchange Man in the Middle



Key Exchange Man in the Middle

- If they do not know each other's public key =>
  - an attacker will be able to successfully break the protocol

Public Key to Exchange Secret Keys



Public Key to Exchange Secret Keys

- Now, our threat model assumes Amy and Bill know each other's public key
- How does Bill know that the symmetric key indeed came from Amy (step 3)?
  - What if a man-in-the-middle attacker replaced the message
    - Other entities may have Bill's public key

Alternative Key Exchange Protocol

- $A \rightarrow B$: $\mathrm{c1} = \mathrm{E}\left(\mathrm{K}_{\mathrm{pub}_\mathrm{B}}, \mathrm{K}_\mathrm{s}\right)$, $\sigma = Sign(K_{priv_A}, c_1)$

K_s = symmetric key

- B: if $(Verify\left(\sigma, K_{pub_A}, c_1\right) == Yes)$
  B: Then: $K_s = D\left(K_{priv_B}, c_1\right)$

Alternative Key Exchange Protocol

- How was K_s calculated?
  - B uses K_{pub_A} to verify signature σ on the ciphertext c
  - B uses k_{priv_B} to decrypt the ciphertext c

Alternative Key Exchange Protocol

- Result: symmetric key has been exchanged
  - Now it can be used for the rest of the communication.
- Why is this protocol more secure?
  - Because the digital signature of A was used
    - Using its private key
  - Only A can sign this

Revised Key Exchange Protocol

- Secure key exchange is complicated
- Many alternative secure protocols exist
  - Protocols need to be proven secure
    - Considering a specific attack model
  - New attacks introduced continuously

Security concepts

- Concepts achieved through symmetric and asymmetric cryptography:
  - Confidentiality: achieved through encryption
    - An eavesdropper can not read message back
  - Authenticity: provided through the digital signature mechanism
  - Non-repudiation: the author of the message can not dispute the original of the message
    - Provided through digital signature

How to Break Cryptography?

- How to break cryptography

Questions?