SJSU SAN JOSÉ STATE UNIVERSITY

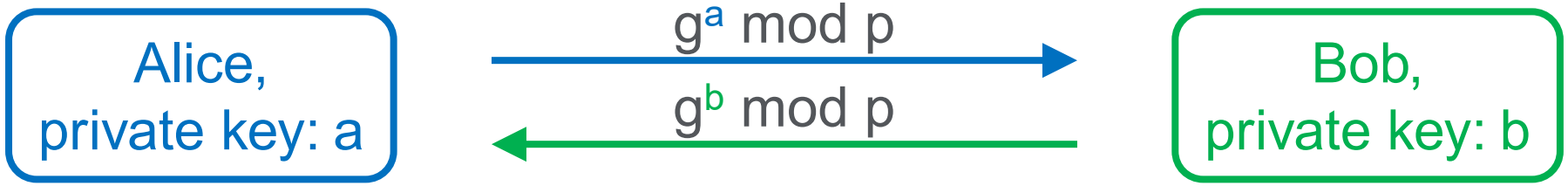
Lesson 7 – Public Key Crypto 3

Yan Chen

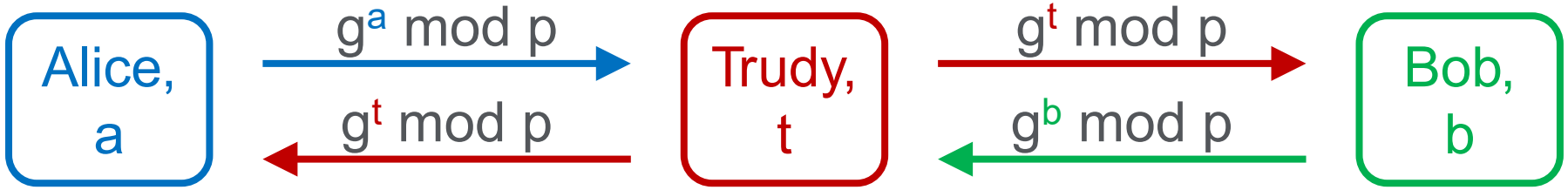
CS166 Fall 2024

Diffie-Hellman

- Diffie-Hellman (DH): key exchanging algorithm
  - Used to exchange symmetric key, NOT encrypt or sign!
- Diffie-Hellman algorithm:

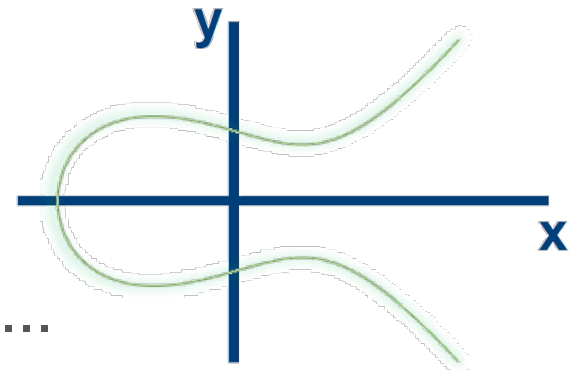


  - ONLY a and b are private
  - After exchange, compute K = g^{ab} mod p as symmetric key
- DH is subject to man-in-the-middle (MiM) attack



ECC

- Elliptic Curve Crypto (ECC): a different way to do the math in public key system using curve $y^2 = x^3 + ax + b$
  - "Elliptic curve" is not a cryptosystem
  - We can have ECC version of DH and RSA, etc.
- Pros: smaller keys, more efficient
  - Roughly speaking, to achieve **same level of security**, the key length for RSA is 10 times the key length for ECC
- Cons: math too complicated
  - No formal proof of security yet
  - Not many people can fully understand it...



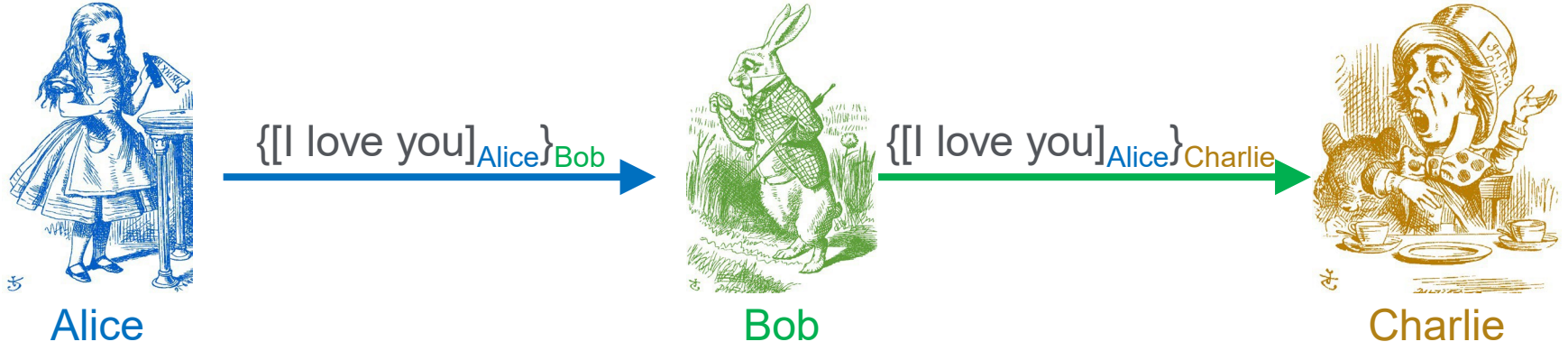
Non-repudiation

- Recall: public key crypto can be also used for signature
  - Sender “encrypts” (sign) message using private key
  - Others verify the signature by “decrypting” using public key
  - That is, public key crypto also provides integrity
- Moreover, this signature also provides non-repudiation
  - Repudiation: signee denies the signature
  - If signed using public key crypto, signee cannot deny after!
  - Assuming private key has not been compromised...

    ONLY the one with the private key can sign the message!
  - Can MAC also provide non-repudiation?

Common Misinterpretations

- Let's see 2 examples first...
- Example 1: sign then encrypt...



- Example 2: encrypt then sign...

Common Misinterpretations

- Misinterpretation 1: signee = sender
  - The signed message $[M]_{Alice}$ is public
  - It is signed by Alice, but others can send $[M]_{Alice}$ to others
  - It's like forwarding a signed paper to others!
  - That is, signee and sender can be different!
- Misinterpretation 2: signee = encrypter = sender
  - Public key is public! So, everyone can compute $\{M\}_{Alice}$!
  - The encrypted message $\{M\}_{Alice}$ is also public
  - Everyone can send $\{M\}_{Alice}$!
  - The signee, the encrypter, the sender all can be different!

- Public Key Infrastructure (PKI): the stuff needed to securely use public key crypto
  - ➢ Generate and manage the keys
  - ➢ Include certificate authority (CA), certificate revocation list, etc.
- CA: a trusted 3rd party (TTP) to create and sign digital certificate for users
- Digital certificate: contains user's name and public key
  - ➢ Also called "public key certificate" or "certificate"
  - ➢ Possibly other info such as birthday, blood type, etc.
  - 💡 Should we minimize the amount of information to include?

- Example: a message M includes Alice's name and her public key: M = (Alice, Alice's public key)
  - CA sign it with its private key: $S = [M]_{CA}$
  - And Alice's Certificate = (M, S)
  - Use CA's public key to verify if $M = \{S\}_{CA}$
- Verify signature to verify integrity & identity of owner of corresponding private key
  - Does NOT verify the identity of the sender of certificatesince certificates are public!
  - Big problem if CA makes a mistake (issue cert. to sb else...)

- PKI can use different “trust models”
  - No general standard for PKI though
- Monopoly model: one CA (“System”)
  - Big problems if CA is ever compromised
- Oligarchy model: “a few” trusted CAs
  - User can decide which CA or CAs to trust
  - This approach is used in browsers today
- Anarchy model: everyone is a CA…
  - Users must decide who to trust
  - This approach used in PGP: “Web of trust”

- Hash Functions
  - Properties of cryptographic hash function
  - The birthday problem
  - MD5 & SHA-1
  - Tiger
  - HMAC

- Public key crypto for digital signature
  - Non-repudiation
  - Common misinterpretations
- Public Key Infrastructure (PKI)
  - Public key certificate
  - Certificate authority
  - PKI trust models: Monopoly, Oligarchy, Anarchy

- Suppose that Bob receives Alice's digital certificate from someone claiming to be Alice.
  - Before Bob verifies the signature on the certificate, what does he know about the identity of the sender of the certificate?
  - How does Bob verify the signature on the certificate and what useful information does Bob gain by verifying the signature?
  - After Bob verifies the signature on the certificate, what does he know about the identity of the sender of the certificate?
- For each situation, better to user MAC or digital signature? Why?
  - Suppose that Alice and Bob want to use a cryptographic integrity check
  - Suppose that Alice and Bob require a cryptographic integrity check and they also require non-repudiation

References

- Stamp, Mark, “Information Security, Principles and Practice, 2nd ed.,” Wiley, New Jersey, USA, 2011