CS 337
Cryptography, HTTPS, Cert Auth

Benjamin Dicken

Cryptography

- ***Cryptography*** (from Greek "kryptós" = "hidden, secret" and "graphein" = "writing") is the practice and study of techniques for secure communication in the presence of third parties called adversaries.
- Constructing and analyzing protocols that prevent third parties or the public from reading private messages
- Relates to ***computer security***:
  - data confidentiality, data integrity

https://en.wikipedia.org/wiki/Cryptography

Cryptography

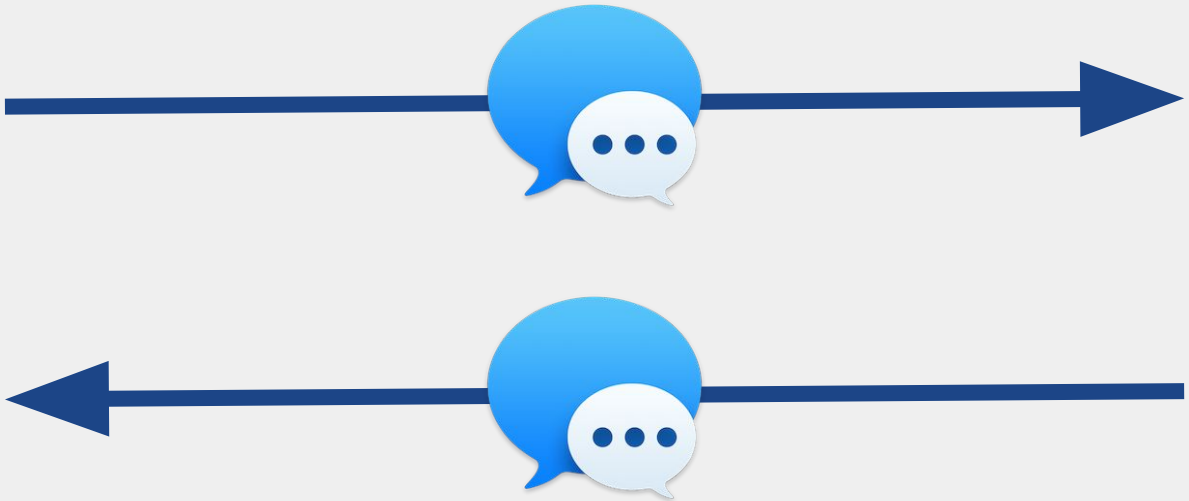
- Modern cryptography exists at the intersection of the disciplines of mathematics, computer science, electrical engineering, communication science, and physics

https://en.wikipedia.org/wiki/Cryptography

Bob

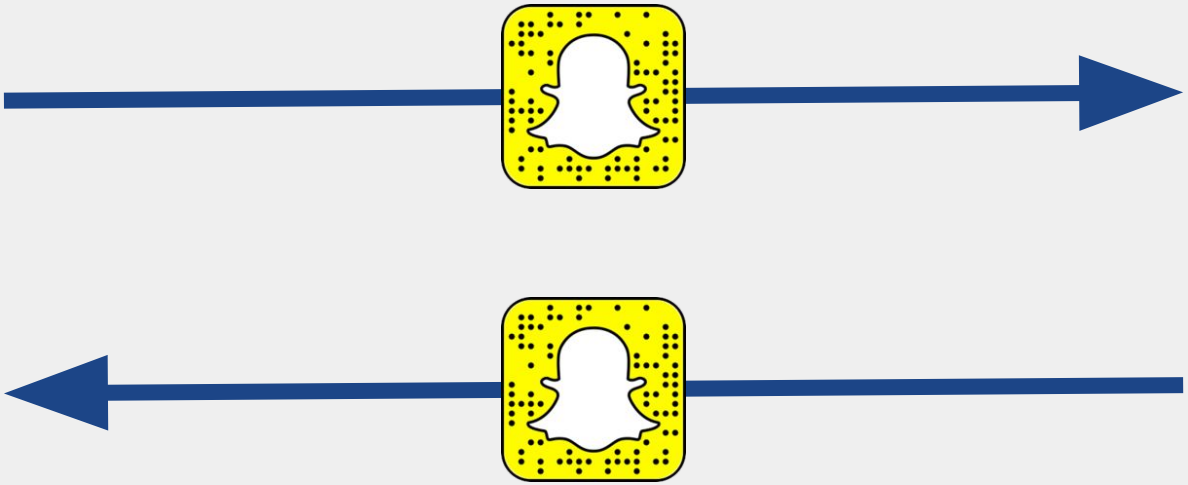
Alice

Bob



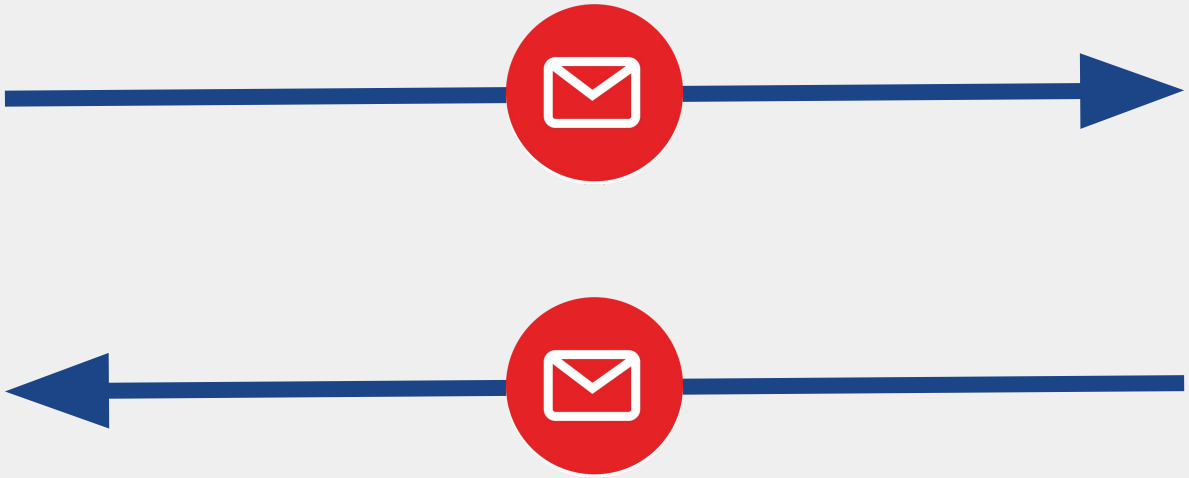
Alice

Bob

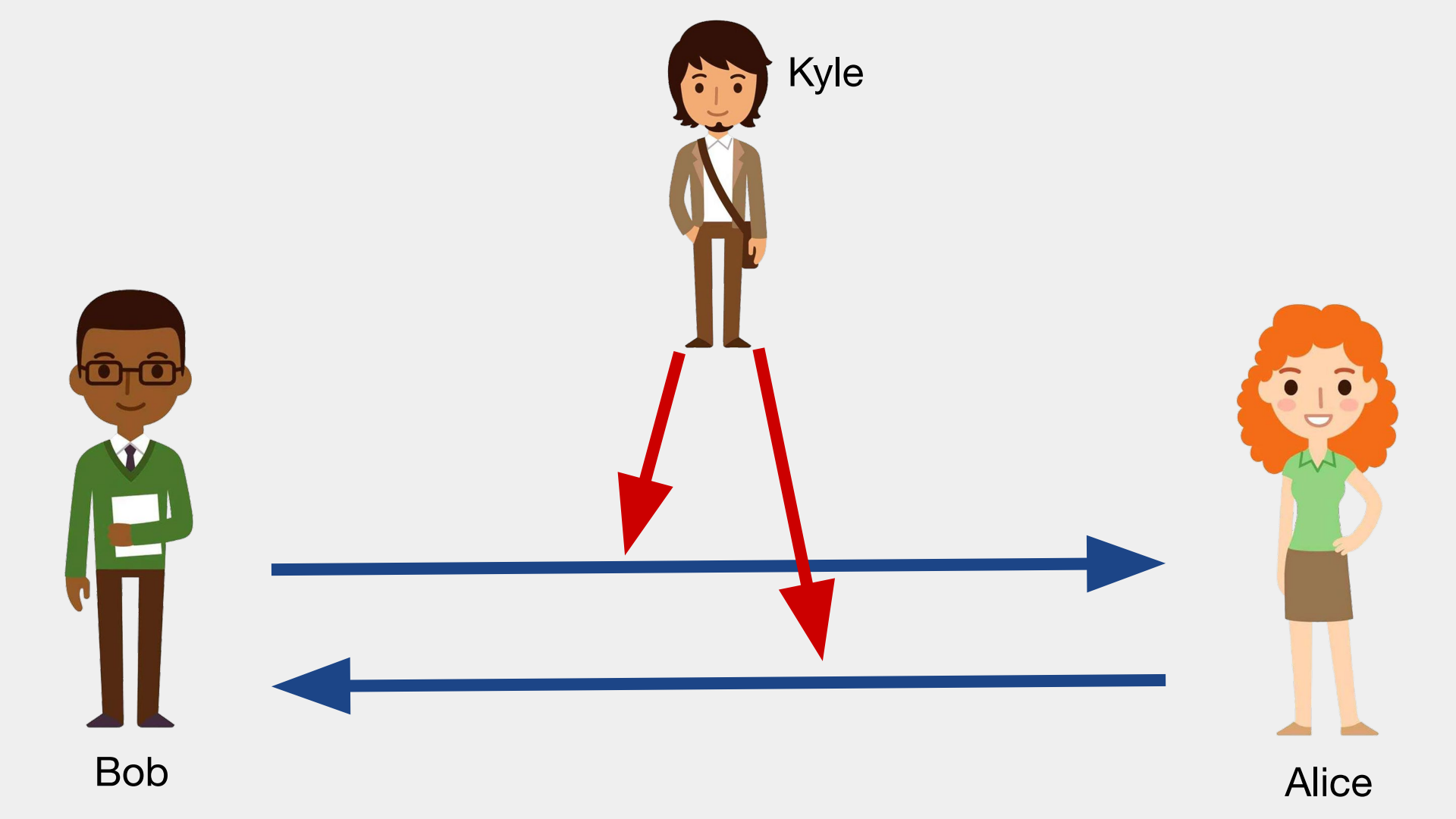


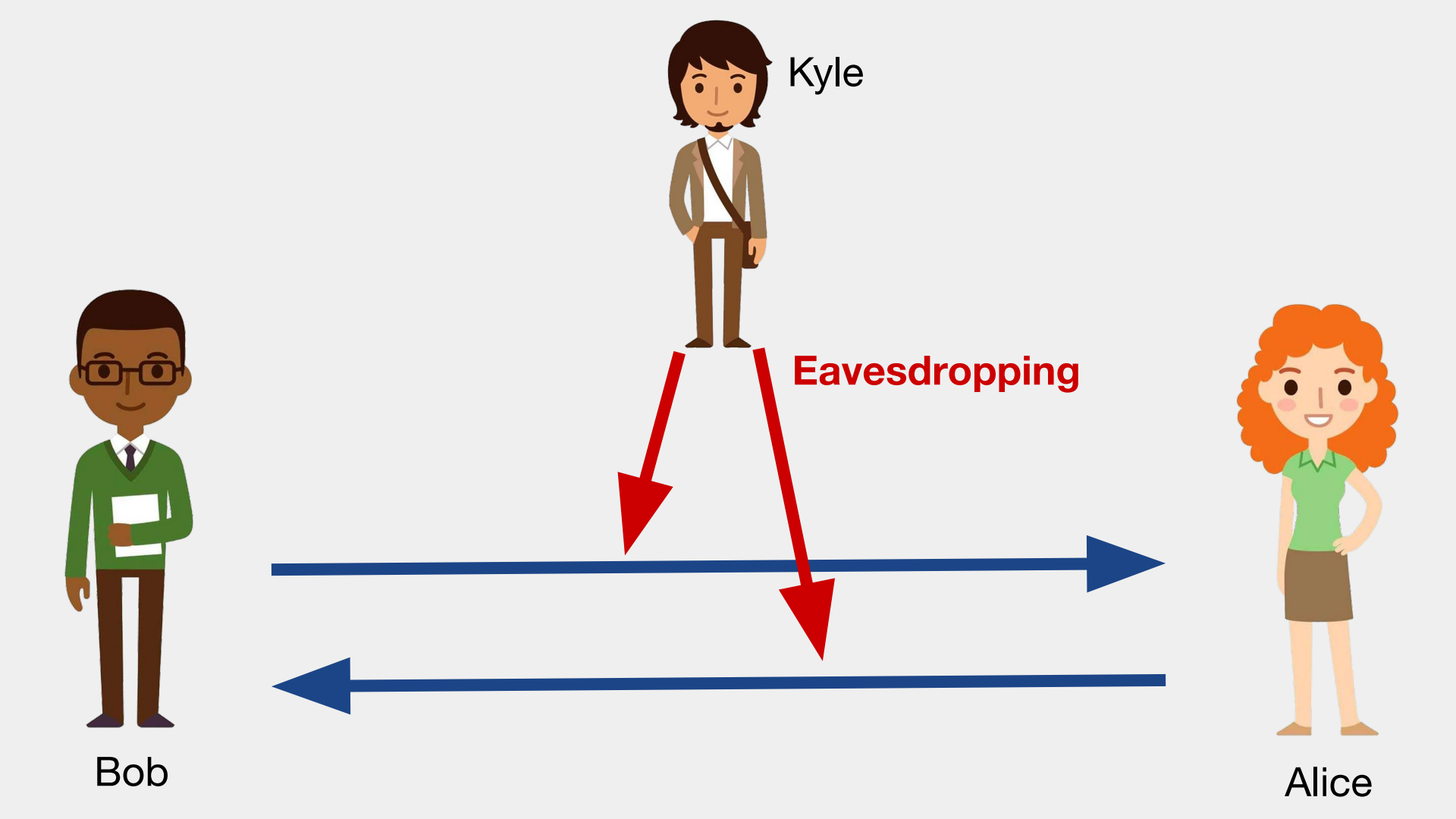
Alice

Bob



Alice


Kyle
Bob
Alice

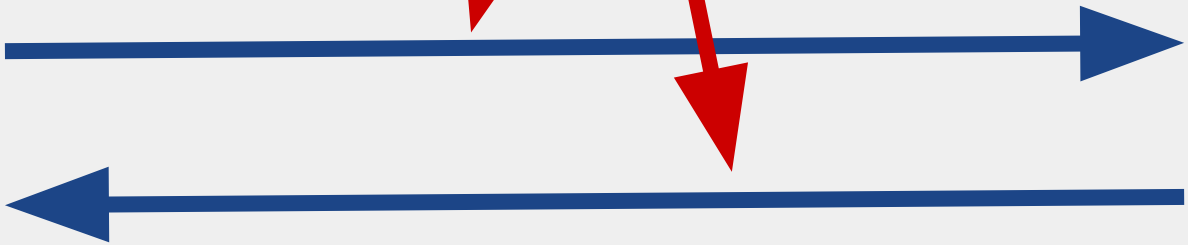

Kyle
Eavesdropping
Bob
Alice

In Cryptography, the idea is to make Eavesdropping ineffective/difficult

Kyle

Eavesdropping

Bob



Alice

In Cryptography, the idea is to make Eavesdropping ineffective/difficult

Kyle

Eavesdropper

Sender/Receiver

Sender/Receiver

Eavesdropping

Bob

Alice

Terminology

- ***Plaintext*** is the text of the original, non-hidden message that the sender desires to communicate to the receiver
- ***Ciphertext*** is the hidden text, that the plaintext is converted to during message transmission
- ***Encrypt***: Plaintext -> Ciphertext conversion
- ***Decrypt***: Ciphertext -> Plaintext conversion

Substitution Cipher

- A ***cipher*** is a way “a secret of disguised way of writing; a code”
- A ***substitution cipher*** is one in which the numbers and letters in a message are replaced by pre-determined other letters/numbers

Substitution Cipher

For example, could use this rule:

To convert plaintext to ciphertext, replace

A->B, B->C, C->D, D->E, . . . X->Y, Y->Z, Z->A

Encrypt the following message: “HOW ARE YOU”

Substitution Cipher

For example, could use this rule:

To ciphertext plaintext to plaintext, replace

A->Z, B->A, C->B, D->C . . .

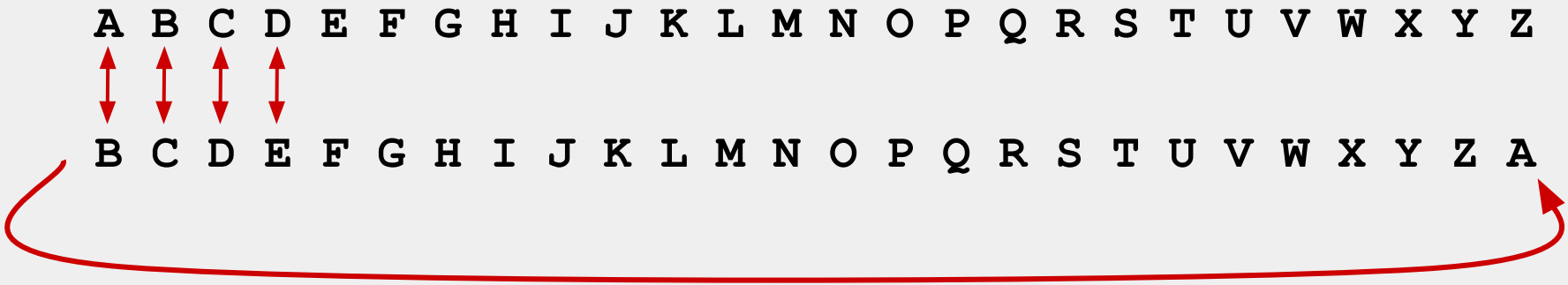
Decrypt the following message: “HPPCZF”

Substitution Cipher

- In the prior example, we substituted each letter with corresponding “shifted” letter using a 1-letter shift
- Specifically called **Caeser cipher**
- Can do this with other shift amounts too

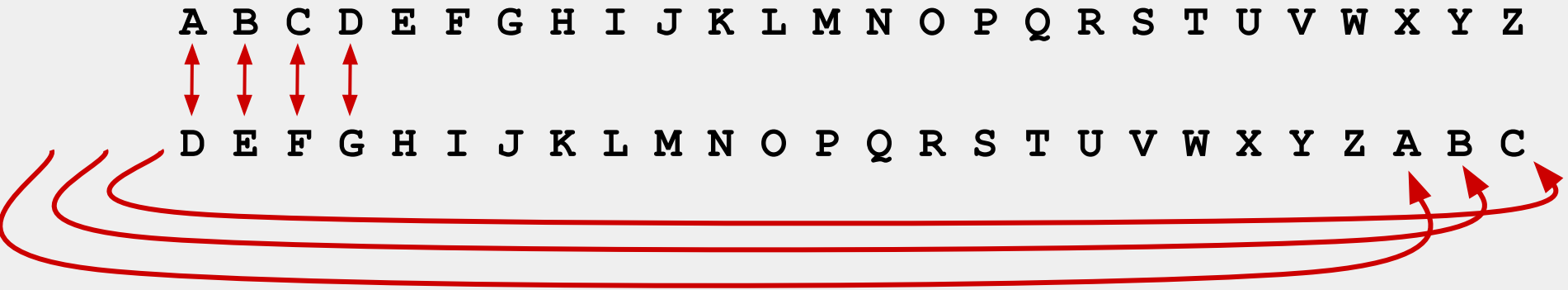
Caesar Cipher 1-Shift

Caesar Cipher 1-Shift



Caesar Cipher 3-Shift

Caesar Cipher 3-Shift



Modern Cryptography

- Substitution ciphers, caesar ciphers, rail-cipher, and others are neat ways of hiding messages
- But they don’t stand much chance against a computer
- Too fast!
- Need more advanced and secure methods of encrypting and decrypting messages

Modern Cryptography

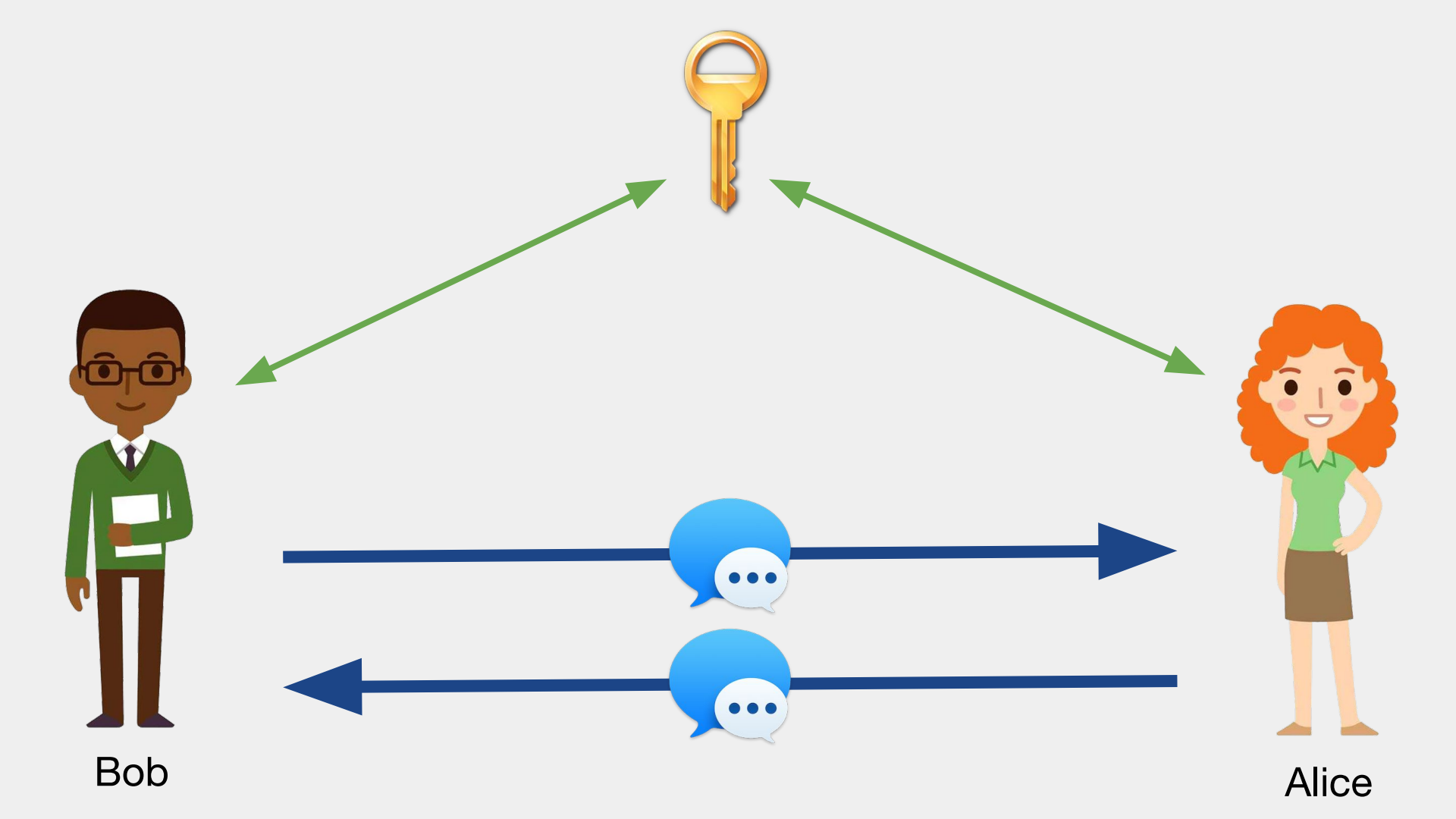
- In most modern protocols, an extra piece of information called a ***key*** is used
- ***Symmetric-key*** and ***Public-key***
- First, let's discuss Symmetric-key

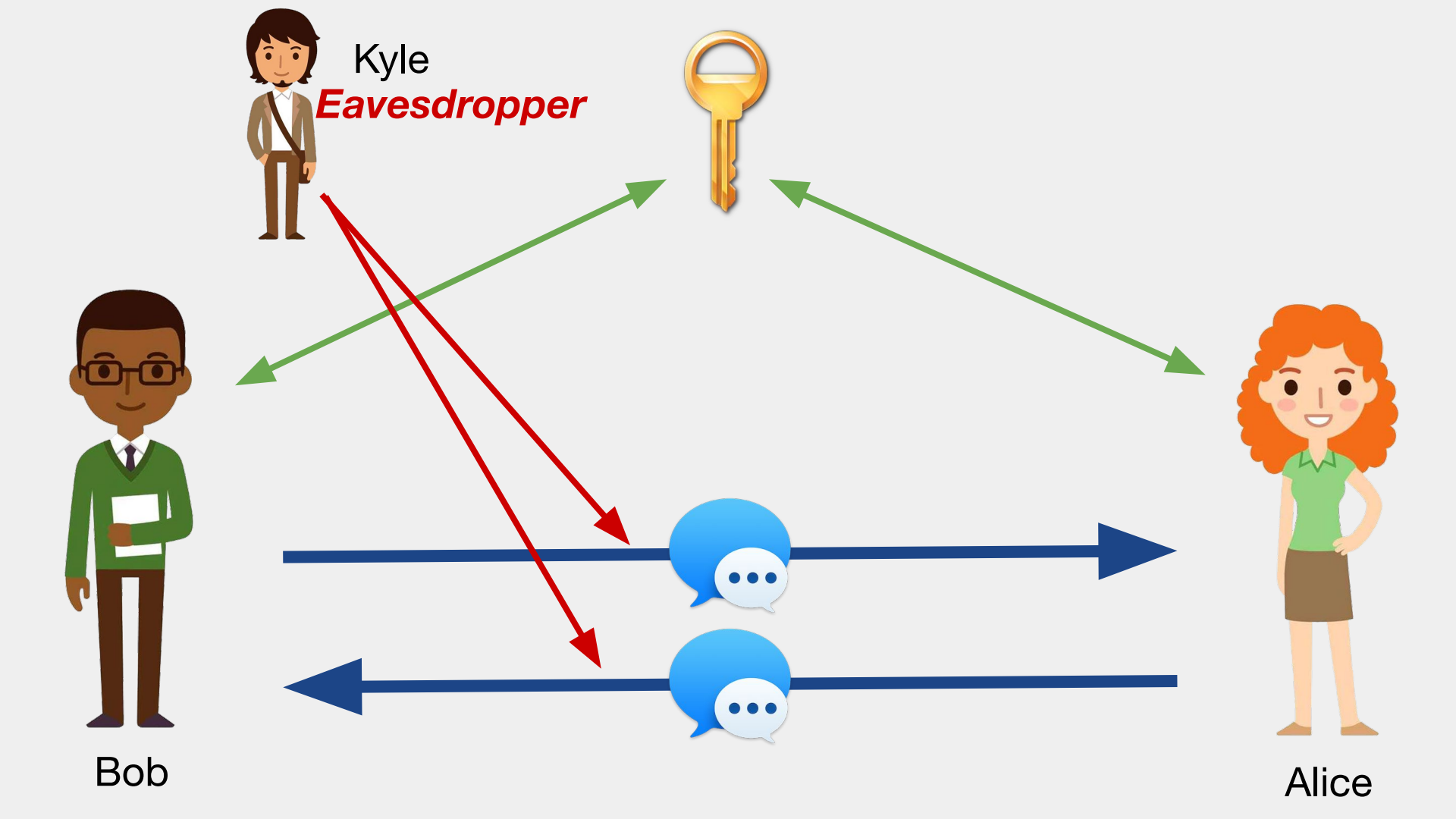
Bob

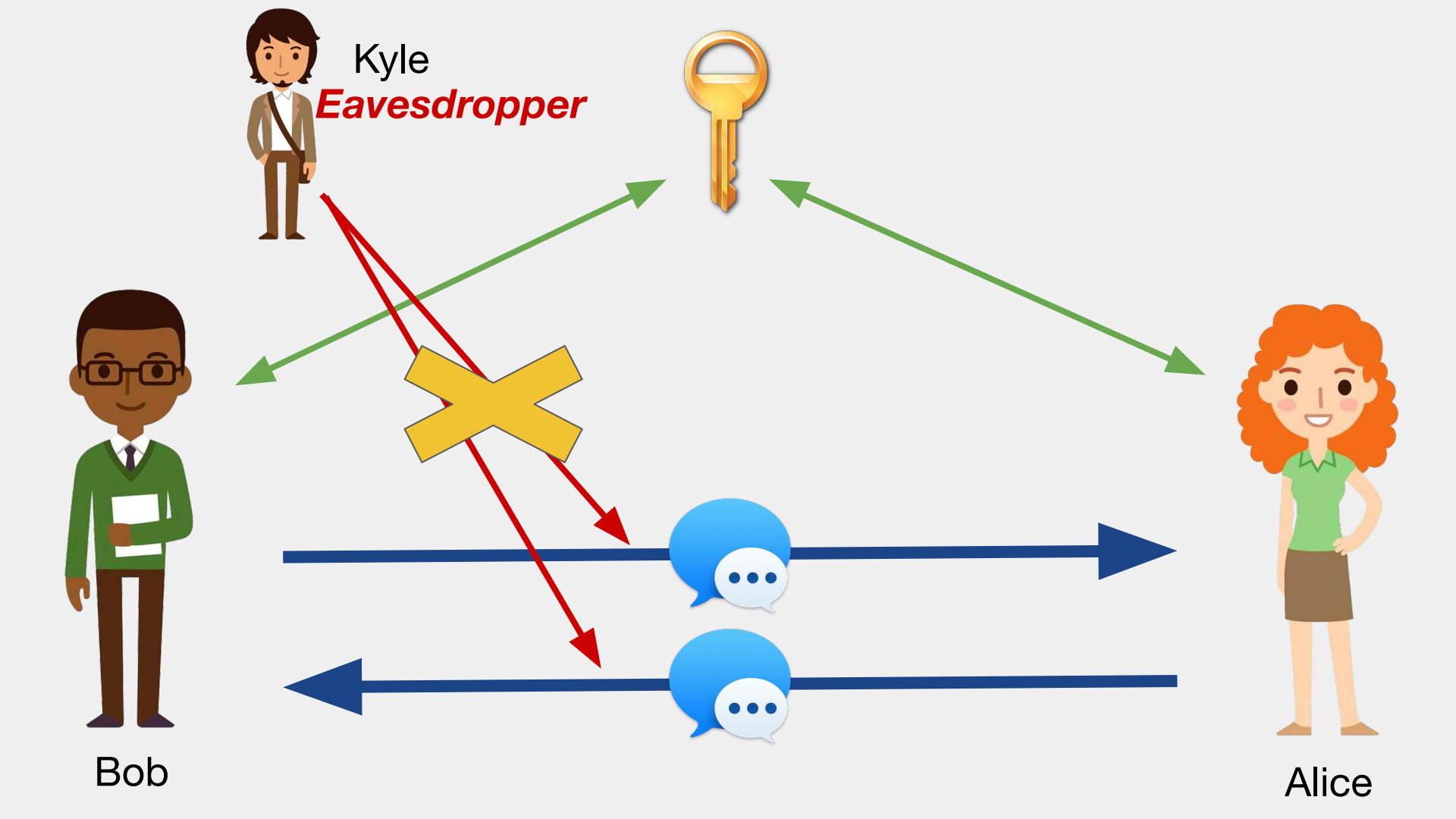
Alice

Bob

Alice


Bob
Alice


Kyle
Eavesdropper
Bob
Alice


Kyle
Eavesdropper
Bob
Alice

Cindy

Bob

Alice

Cindy

Bob and Alice
share a key

Bob

Alice

Cindy

Bob

Bob sends message to Alice

Alice

Cindy

Good!

Bob

Bob sends message to Alice

Alice

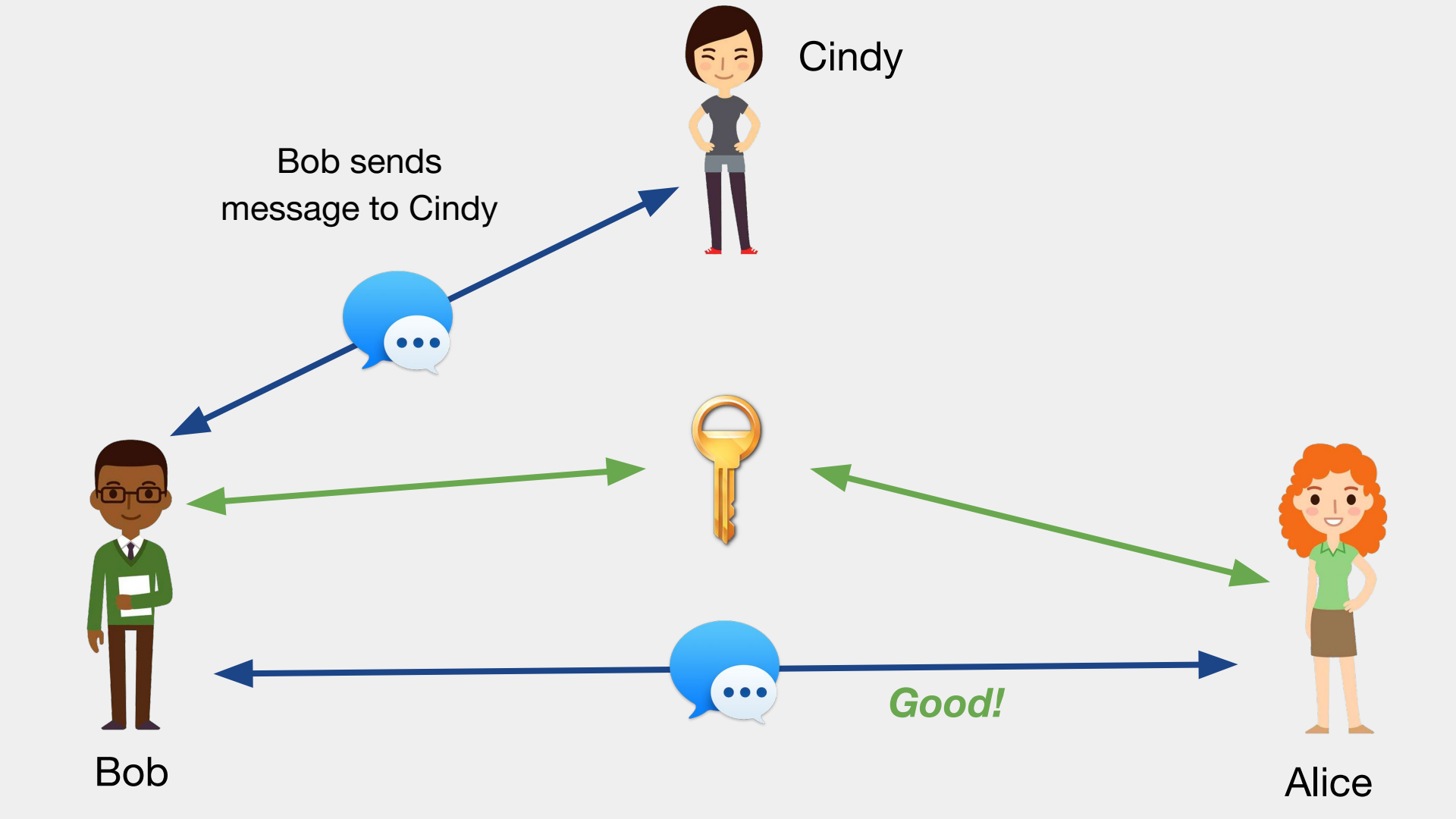
Cindy

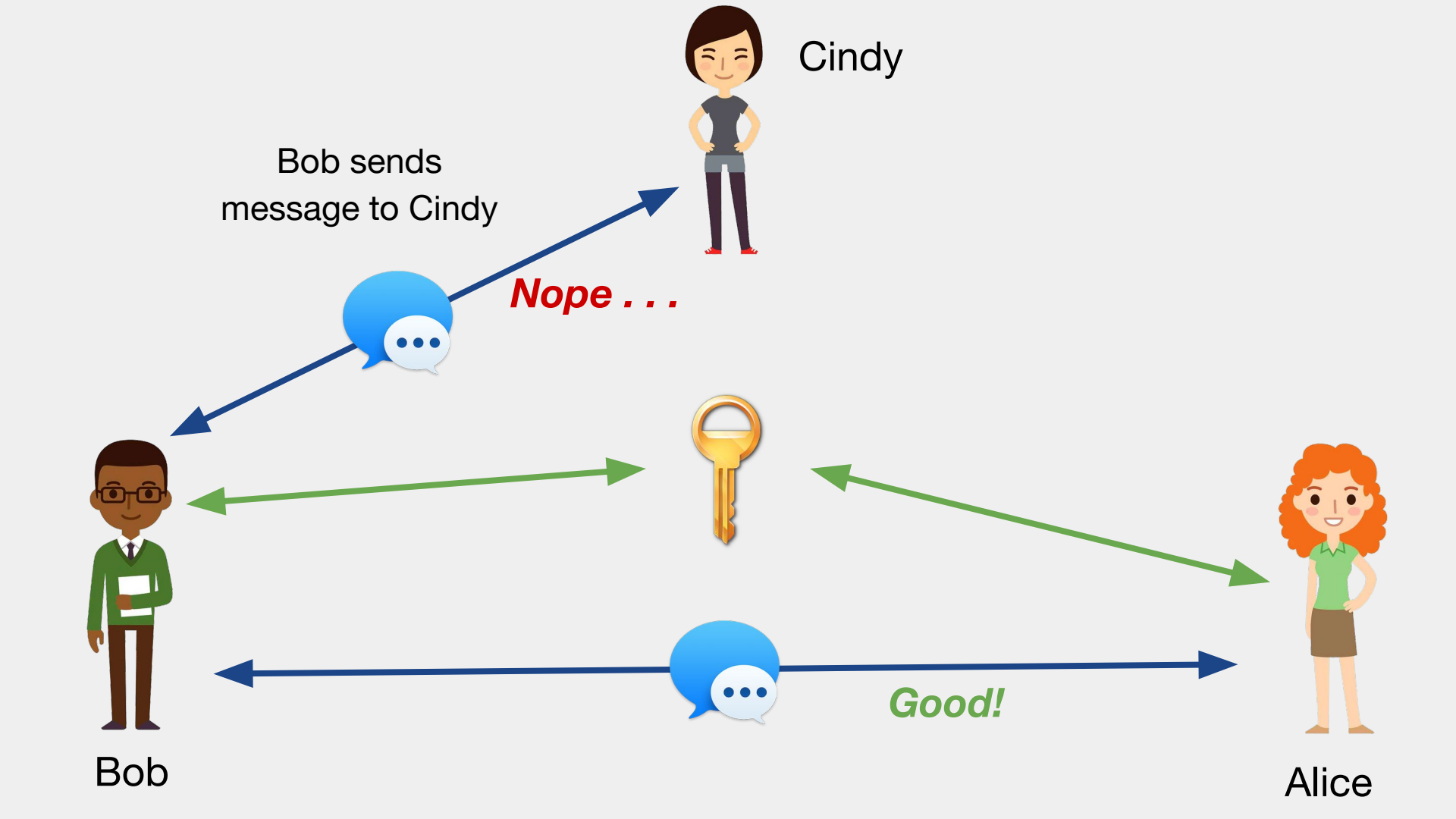
Good!

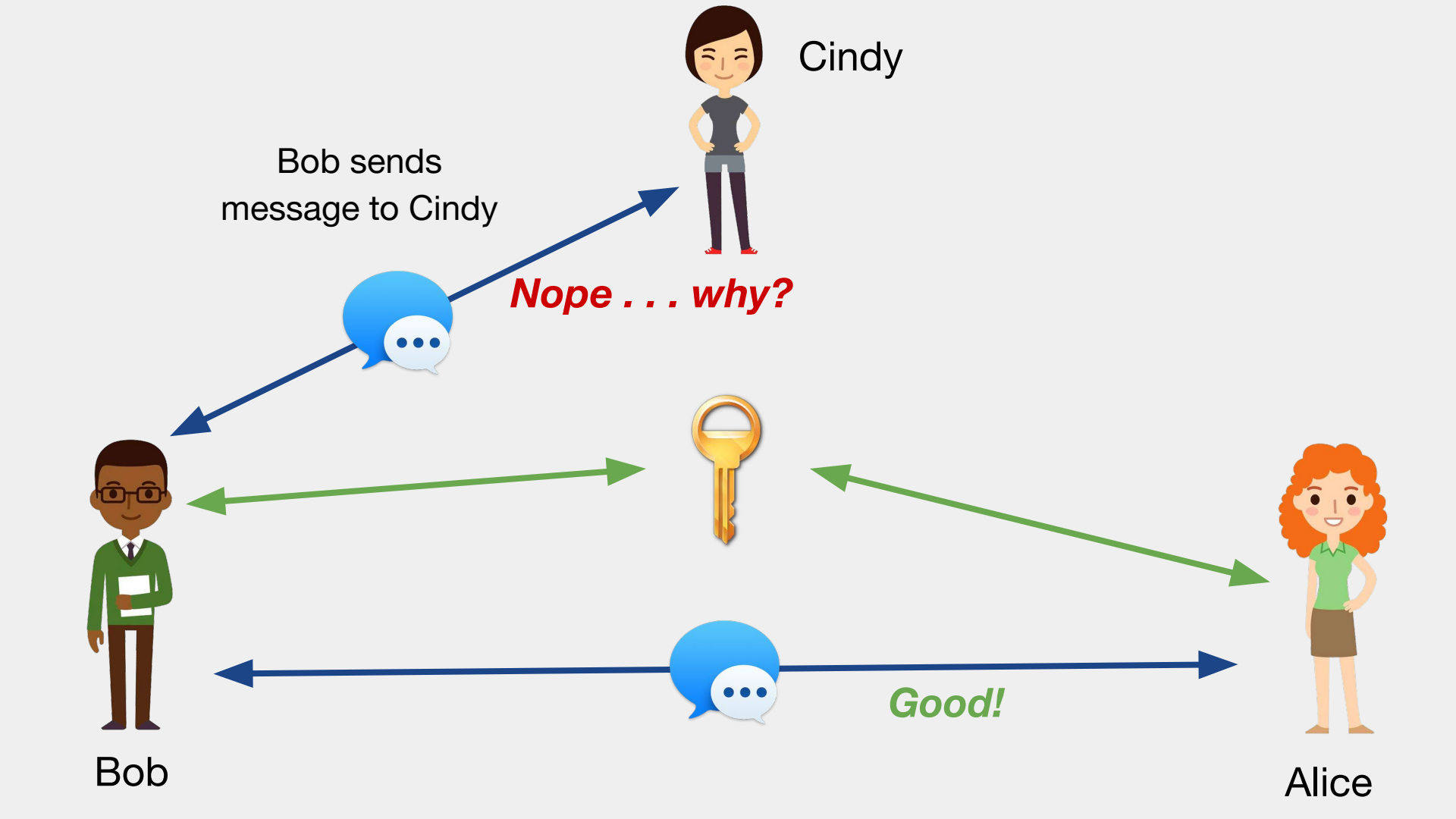
Bob

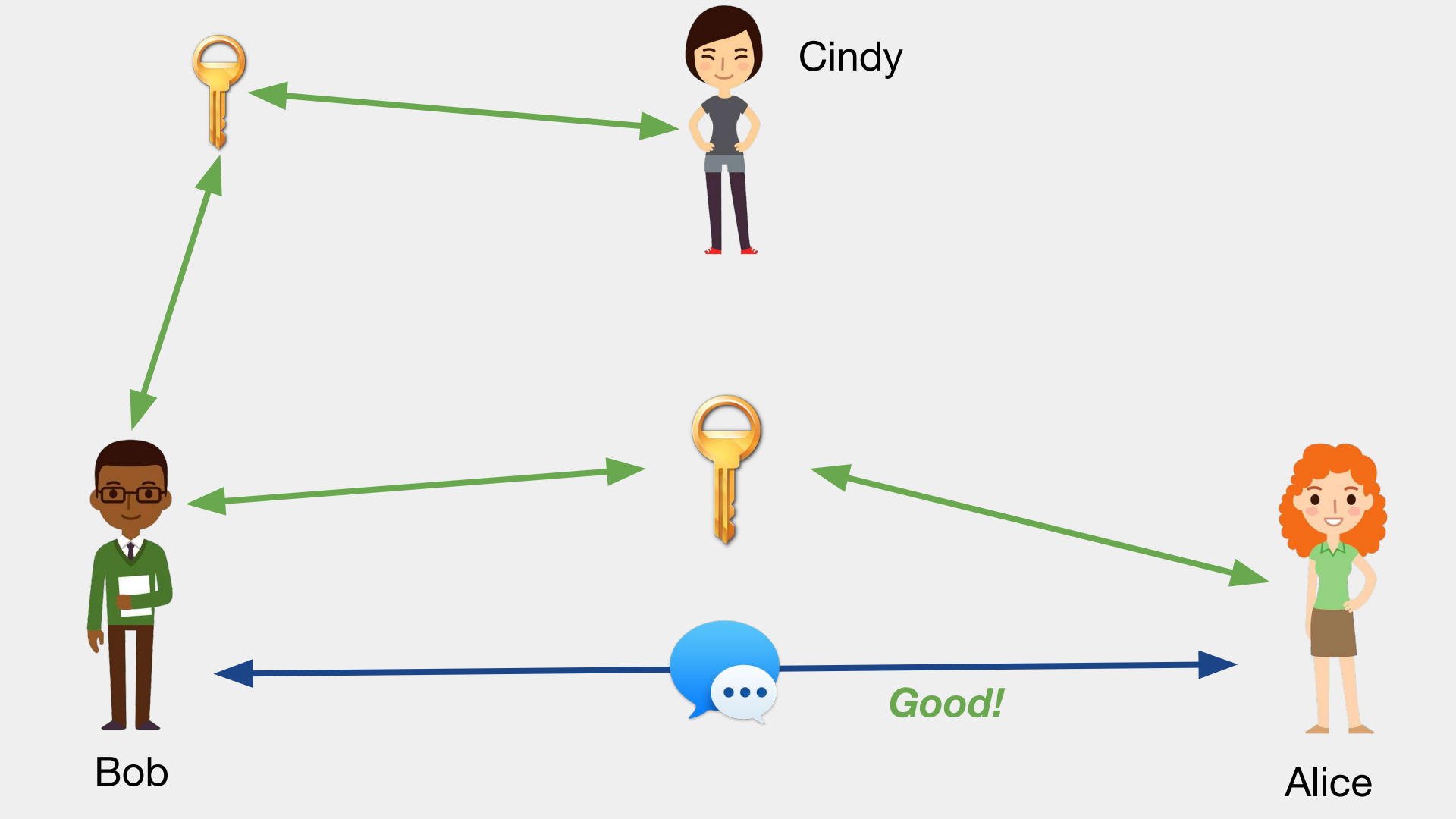
Can send messages in both directions

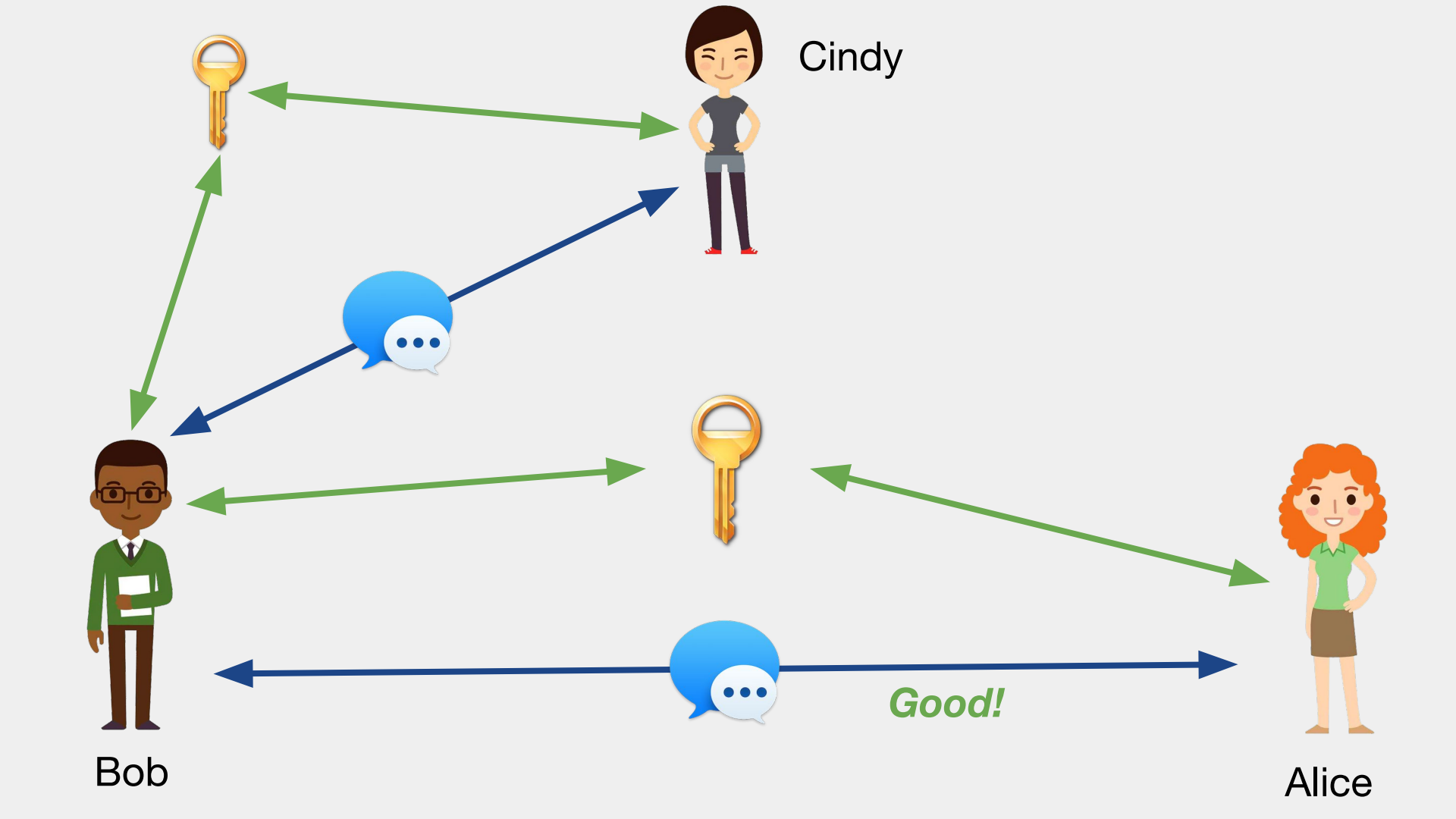
Alice

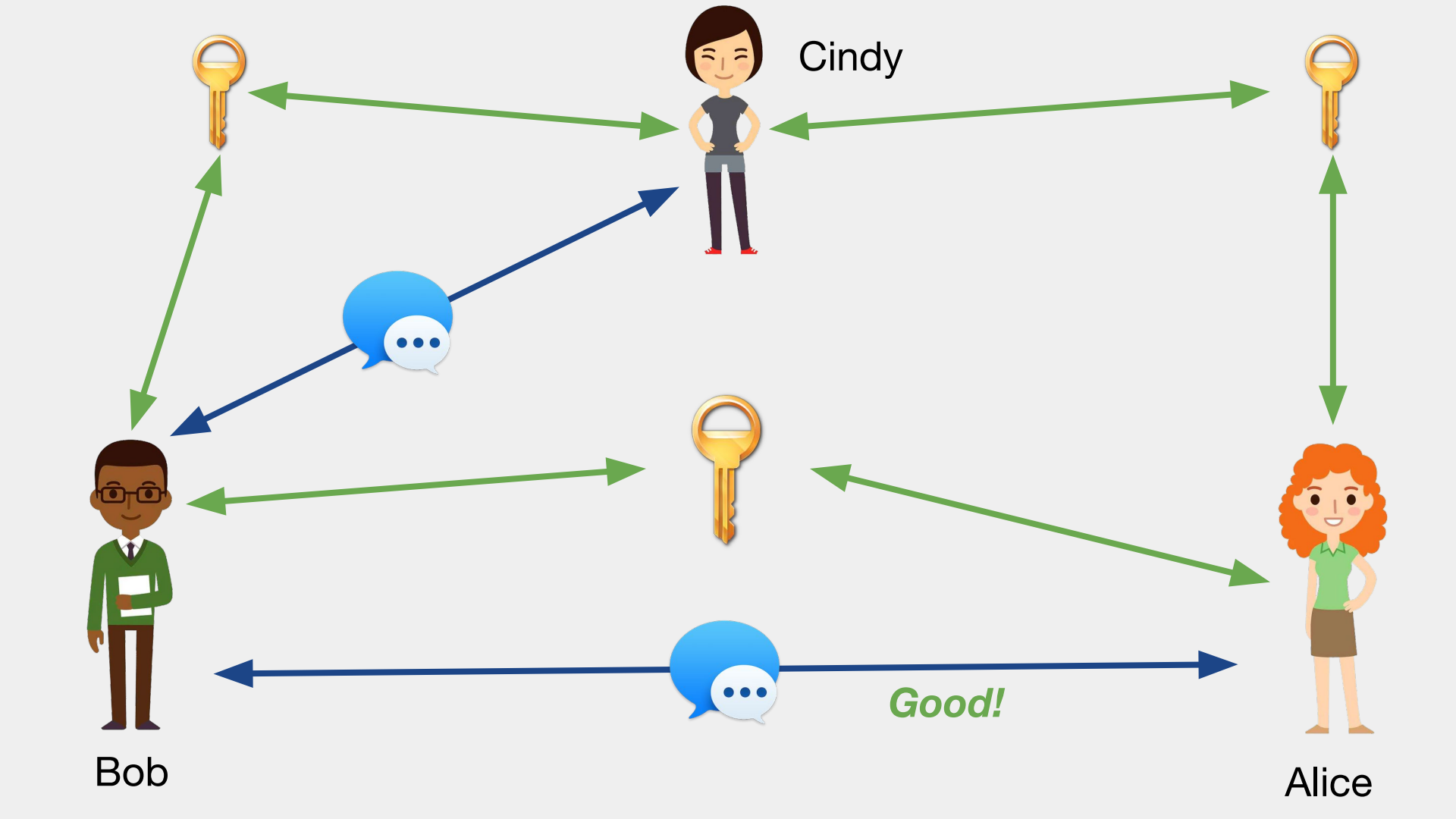

Cindy
Bob sends
message to Cindy
Good!
Bob
Alice

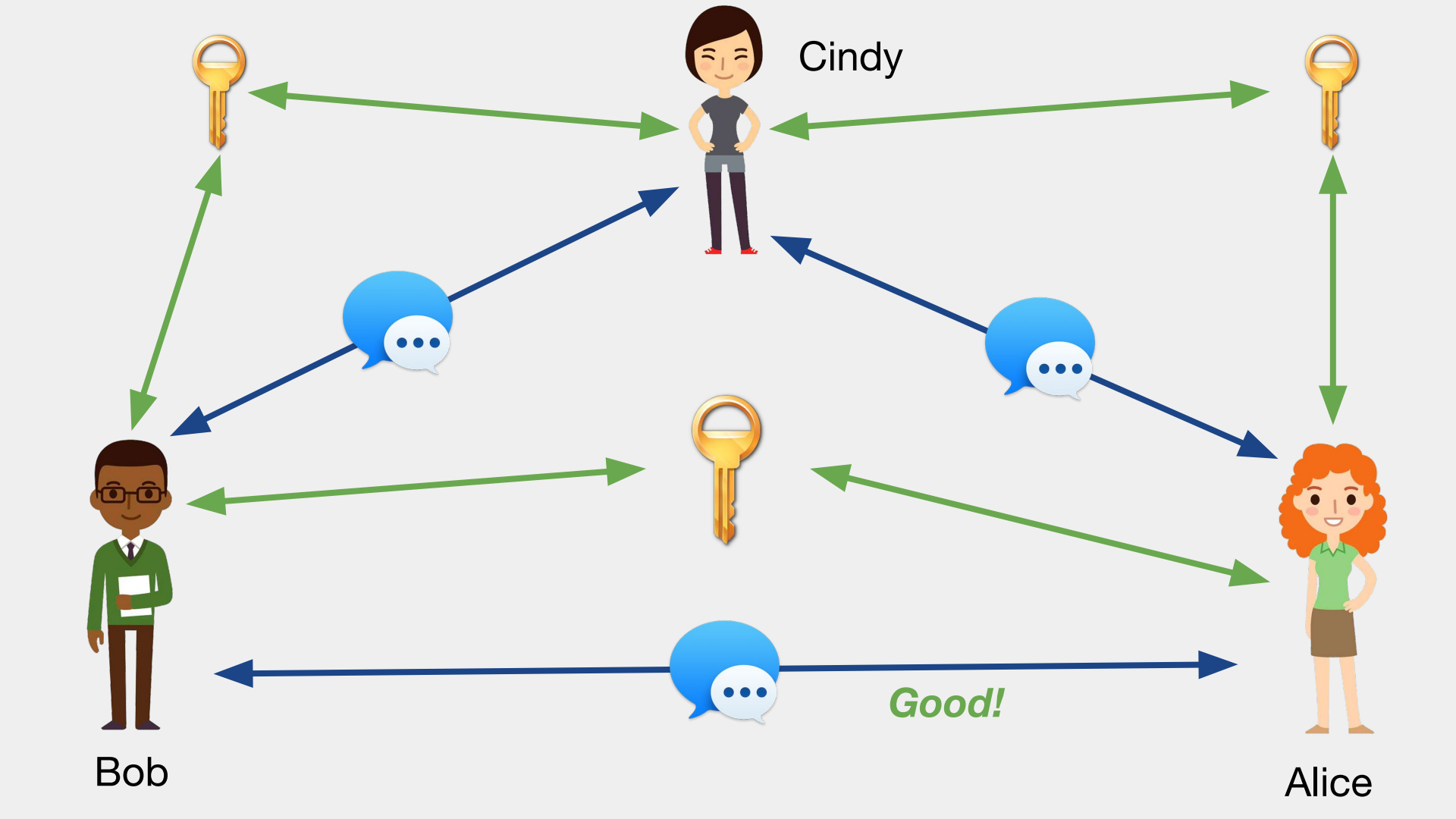

Cindy
Bob sends
message to Cindy
Nope . . .
Good!
Bob
Alice


Cindy
Bob sends
message to Cindy
Nope . . . why?
Good!
Bob
Alice


Cindy
Good!
Bob
Alice

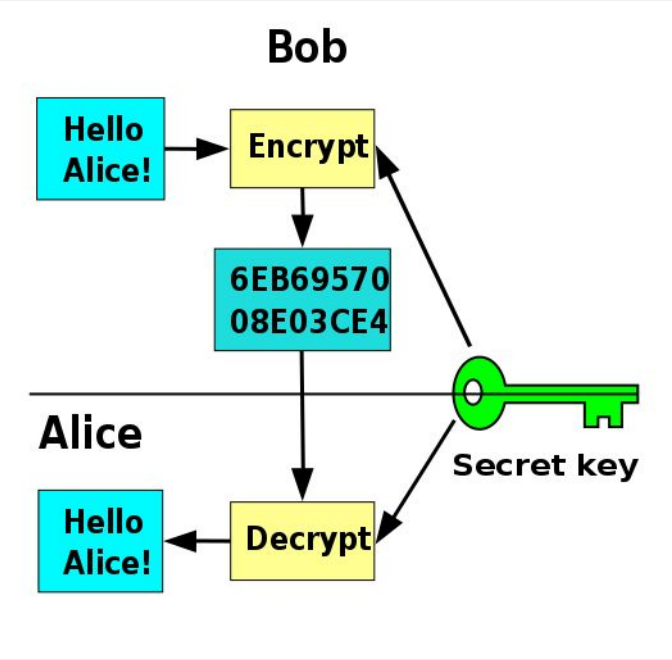

Cindy
Good!
Bob
Alice


Cindy
Good!
Bob
Alice


Cindy
Good!
Bob
Alice

Symmetric-key Cryptography

- In ***Symmetric-key*** cryptography, both “parties” in a communication transaction share an identical key
- Digital key
- Typically, this is some long sequence of text, or large number, which is needed to decrypt a message
- Uses the same key for encryption and decryption



Sam

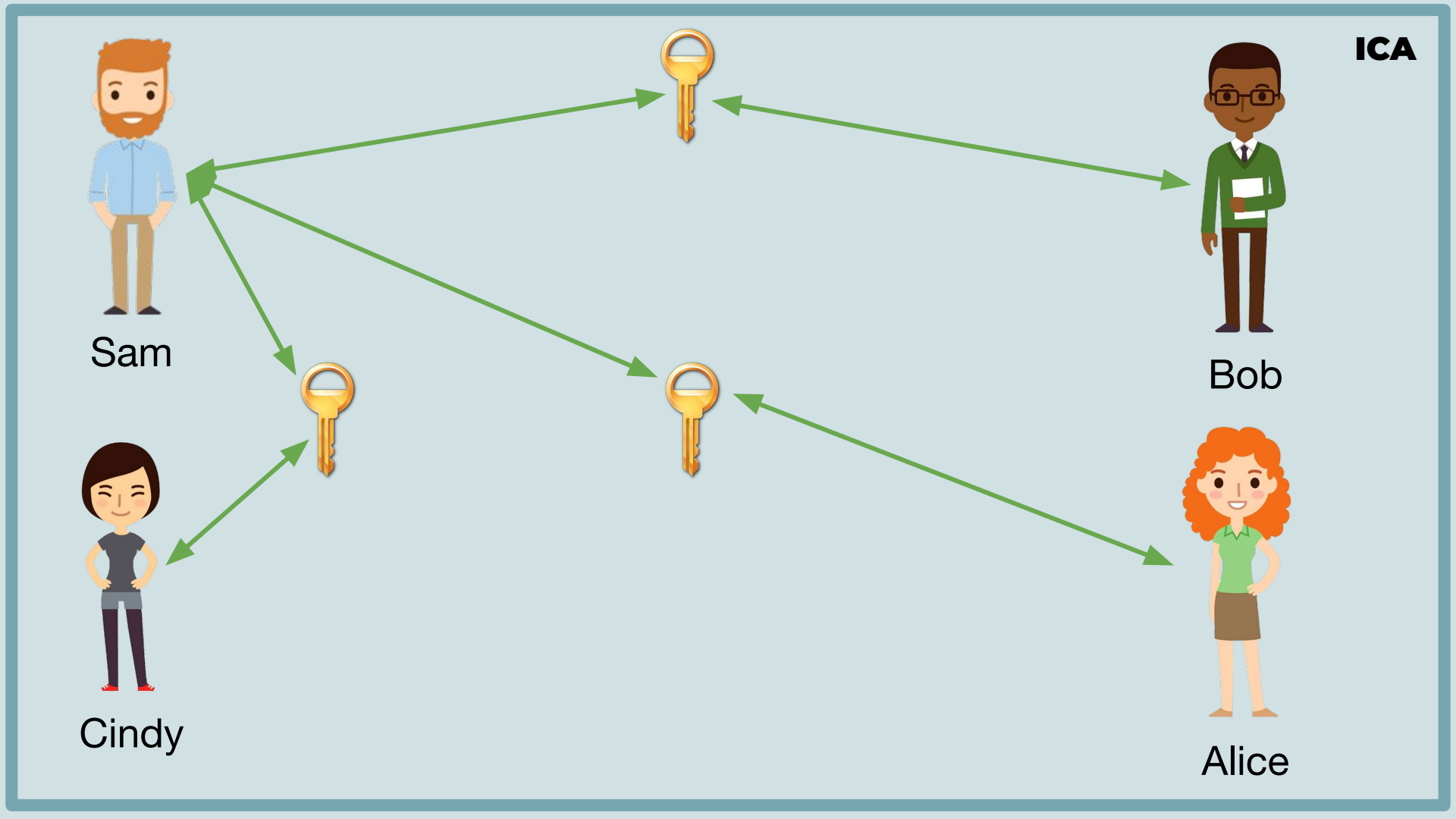
Cindy

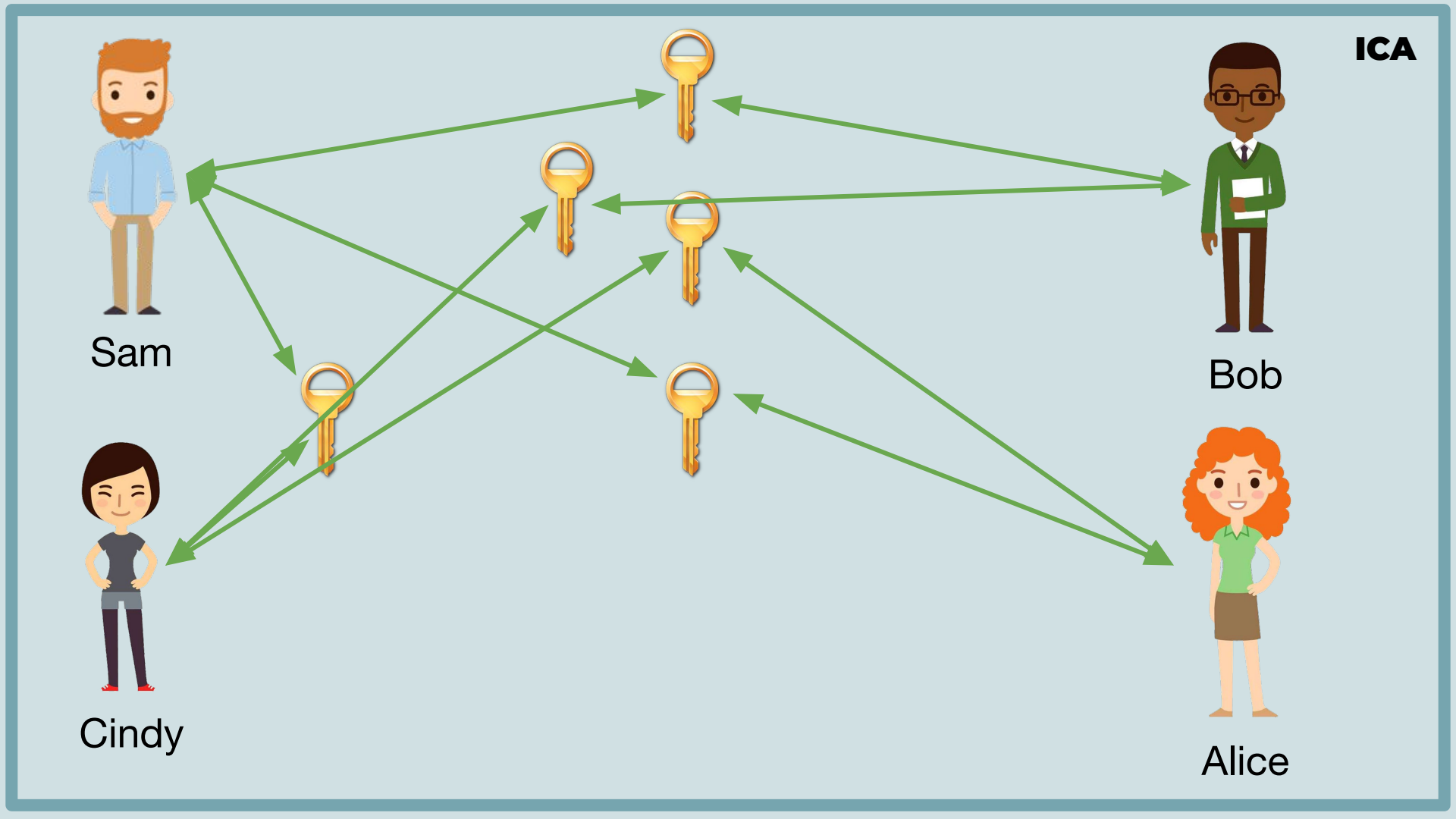
How many keys?

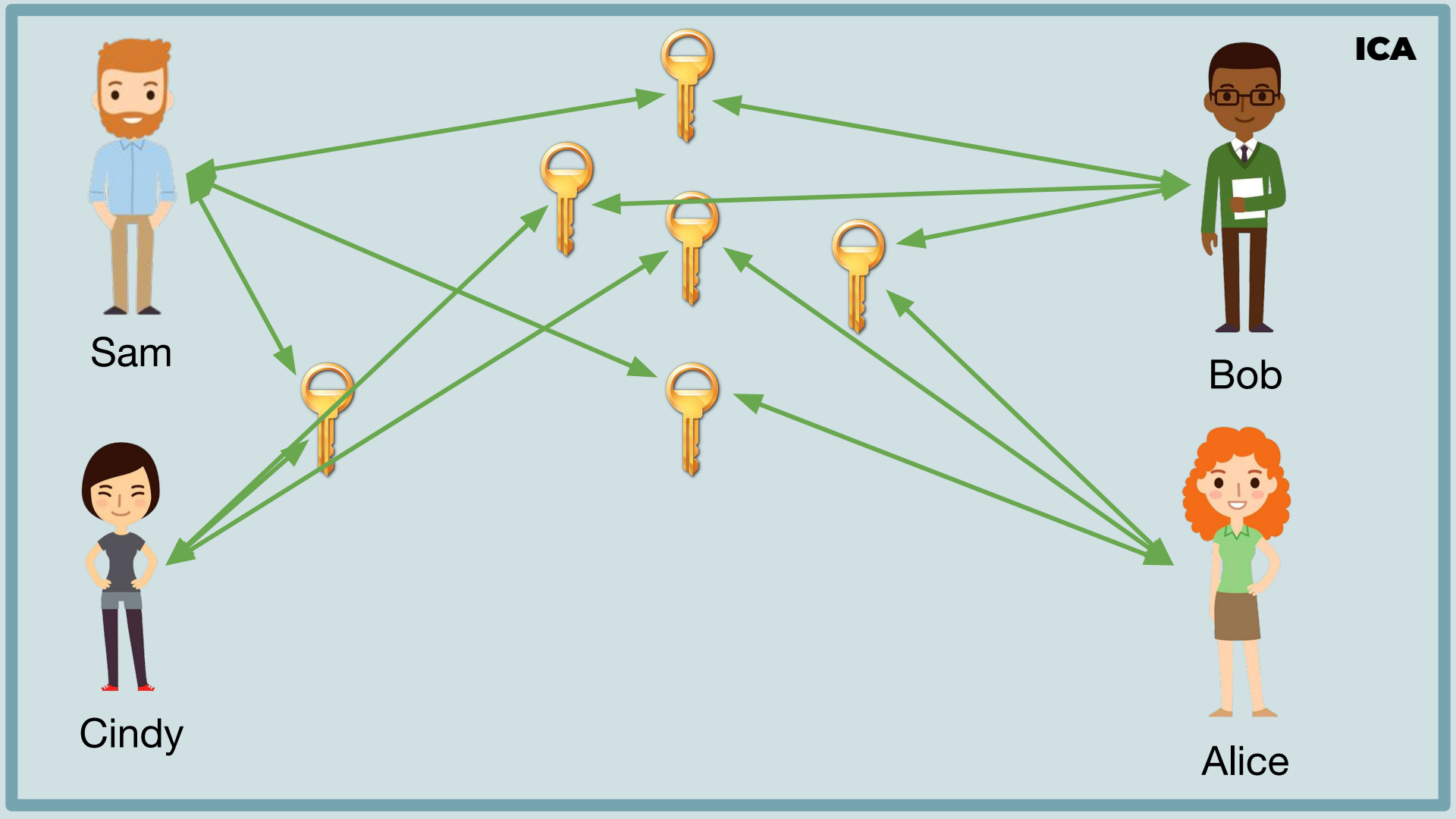
Each person should be able to send *individual* messages to the other

Bob

Alice


ICA
Sam
Bob
Cindy
Alice


ICA
Sam
Bob
Cindy
Alice


ICA
Sam
Bob
Cindy
Alice

ICA

How many keys?

Each person should be able to send *individual* messages to the other

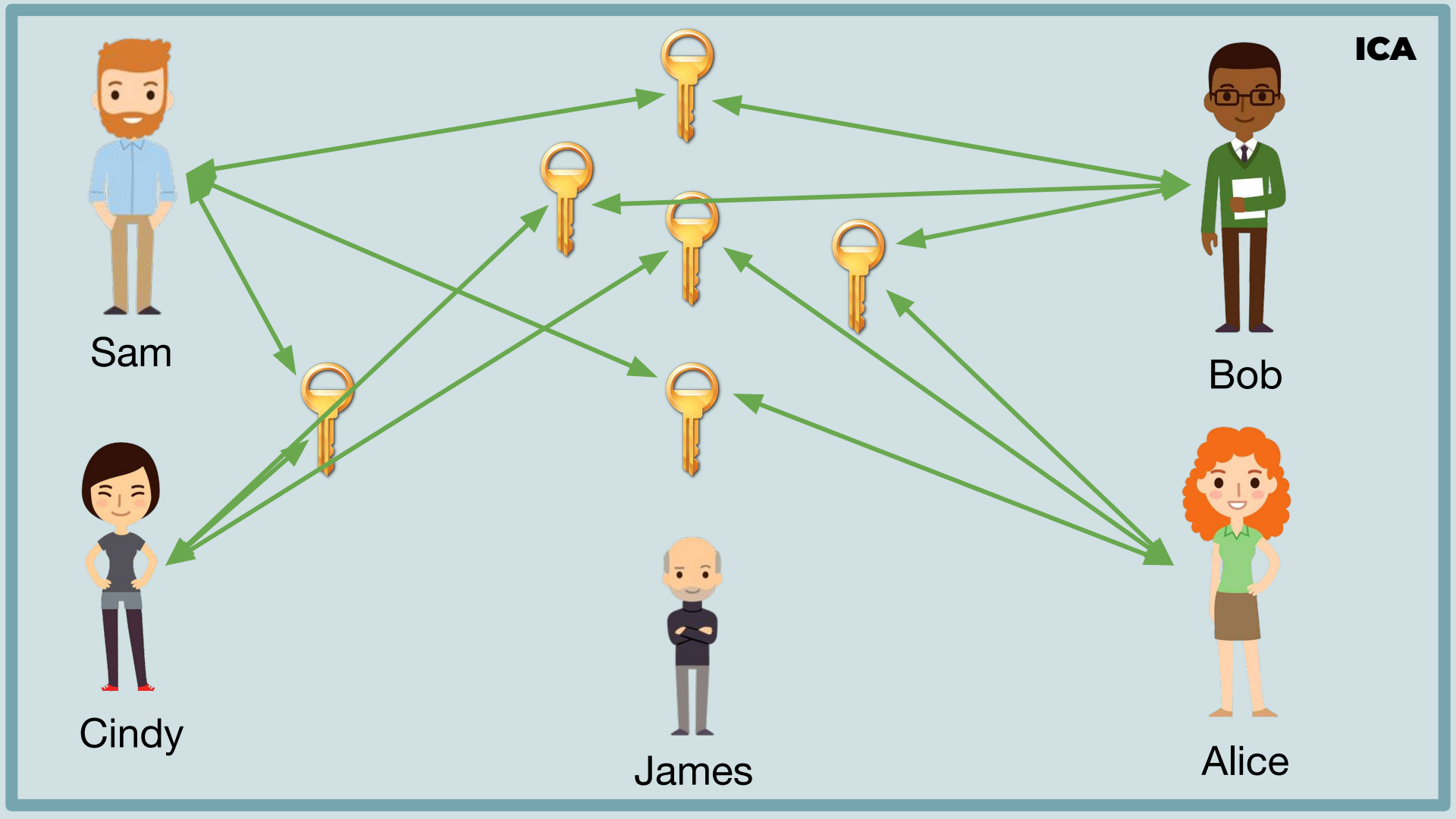
Sam

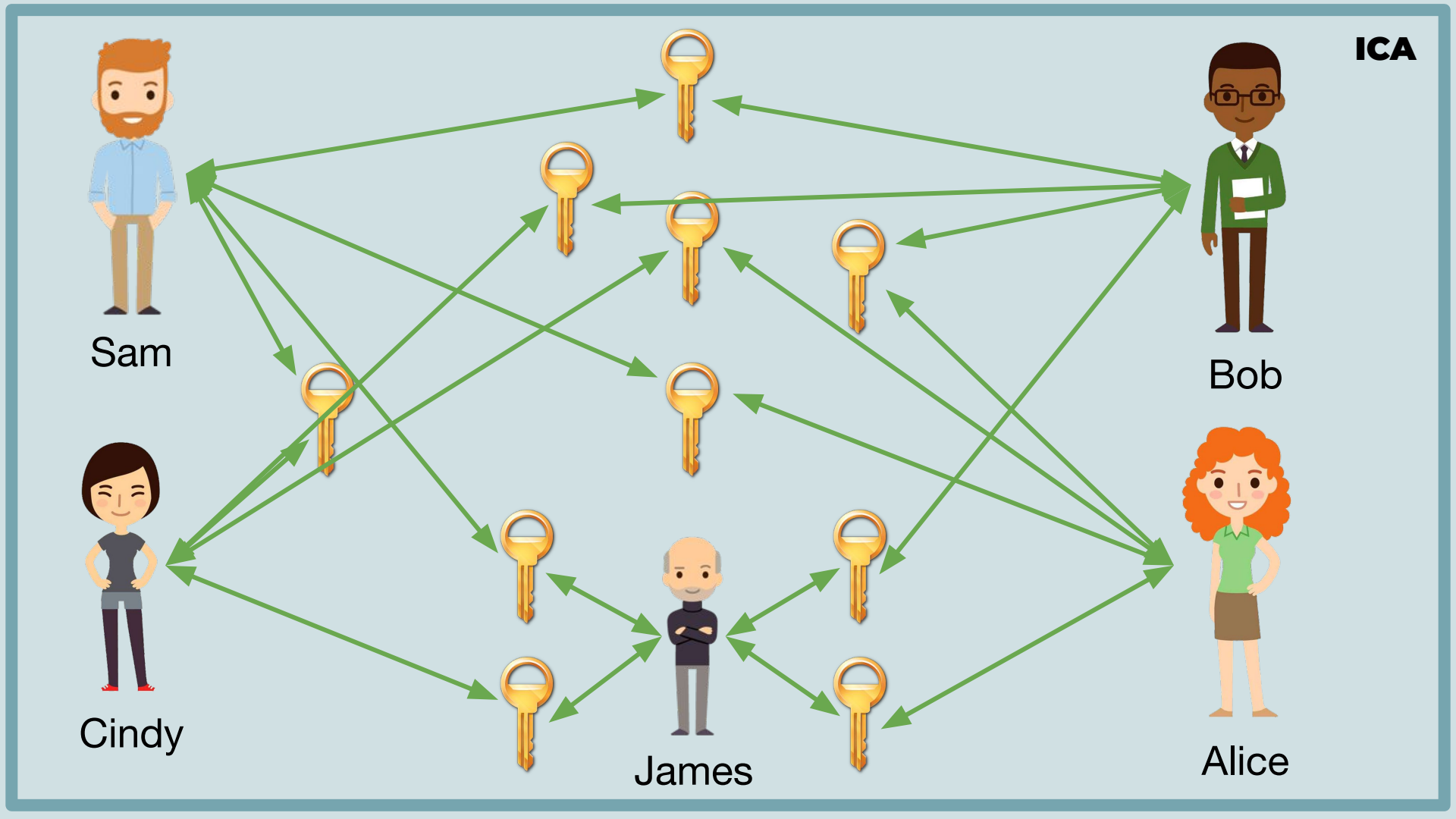
Bob

Cindy

James

Alice


ICA
Sam
Bob
Cindy
James
Alice

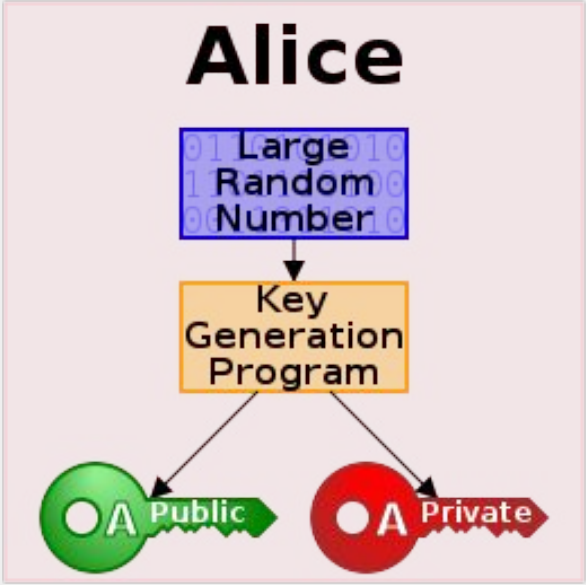

ICA
Sam
Bob
Cindy
James
Alice

Too many keys!

- Symmetric key downside: Unique key for all pairs of people communicating!
- Many keys

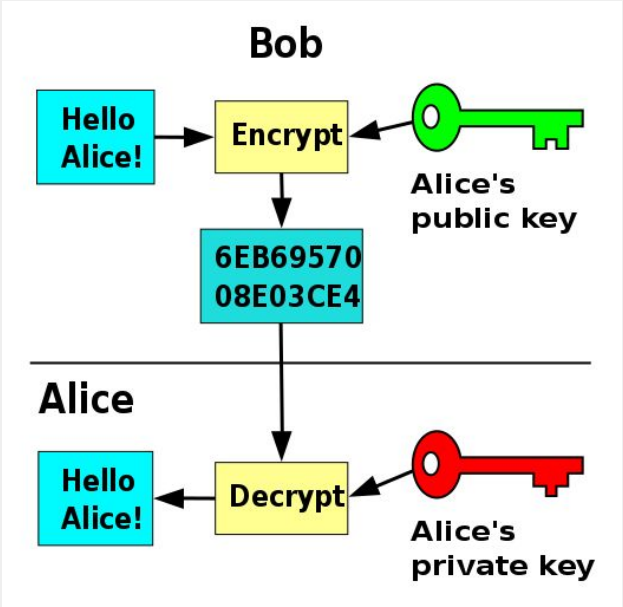
Public-key Cryptography

- The solution is ***Public-key cryptography***
- An entity X needing to communicate generates a pair of keys - the **public key** and **private key**
- Each user wanting to communicate must generate both version of this key
- Then, the user makes the public key available to anyone who they want to receive messages from, and keeps private key to themselves



Public-key Cryptography

- The solution is ***Public-key cryptography***
- An entity X needing to communicate generates a pair of keys - the **public key** and **private key**
- Messages sent to X are encrypted with the public key (available to many) and decrypted with X's private key (only available to X)



Cindy

Bob

Alice

Cindy

Alice generates public and private key. Makes pubic available to all, keeps private secret

Bob

Alice

Cindy

Alice generates public and private key. Makes pubic available to all, keeps private secret

Bob

Alice

Cindy

Bob encrypts message to alice with her public key, send to alice, only she can decrypt

Encrypt

Bob

Alice

Cindy

Bob encrypts message to alice with her public key, send to alice, only she can decrypt

Bob

Encrypt

Decrypt!

Alice

Cindy can also send messages to alice, using the same key and same process as Bob

Cindy

Encrypt

Bob

Alice

Cindy can also send messages to alice, using the same key and same process as Bob

Cindy

Encrypt

Bob

Decrypt!

Alice

Cindy and Bob cannot eavesdrop on each-others messages to alice, because only she can decrypt

Cindy

Encrypt

Encrypt

Bob

Decrypt!

Alice

Cindy

Bob also generates public and private key. Makes pubic available to all, keeps private secret

Bob

Alice

Cindy

Bob also generates public and private key. Makes pubic available to all, keeps private secret

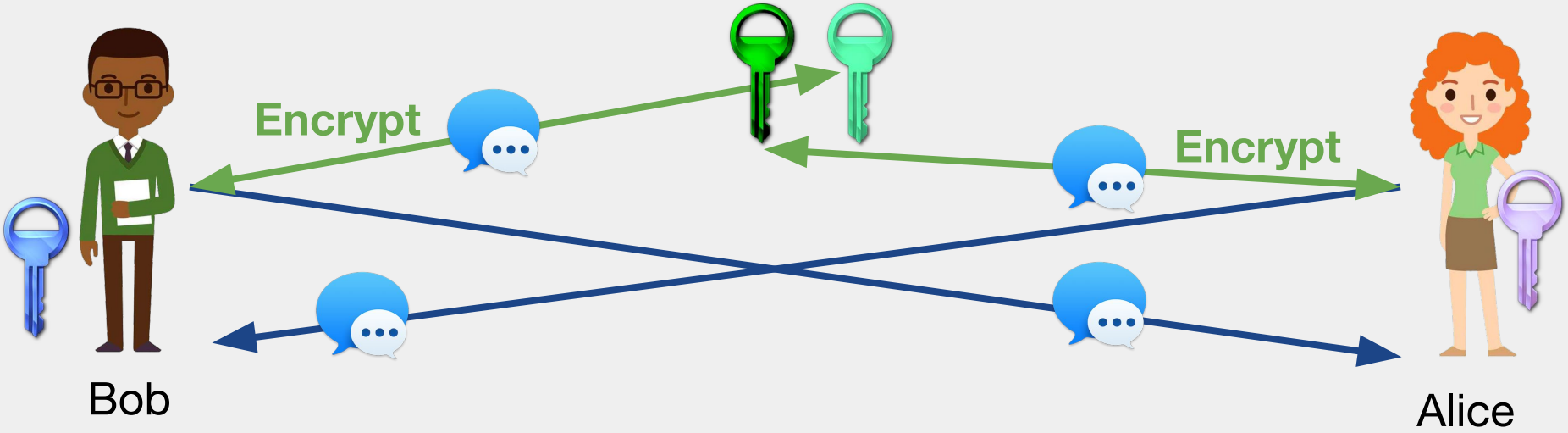
Bob

Alice

Cindy

Now, bob and alice can communicate back-and-forth

Anyone with access to their public keys can send them messages



ICA

How many keys?

Each person should be able to send *individual* messages to the other

Sam

Bob

Cindy

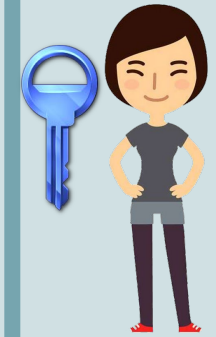
James

Alice

ICA

Sam

Bob



Cindy

James

Alice

Comparing pub-key and symmetric key

For **N** people to send each-other private messages,
how many keys are needed for

- Symmetric Key: $1 + 2 + \ldots + \mathbf{N}$
- Public-key: $2 * \mathbf{N}$

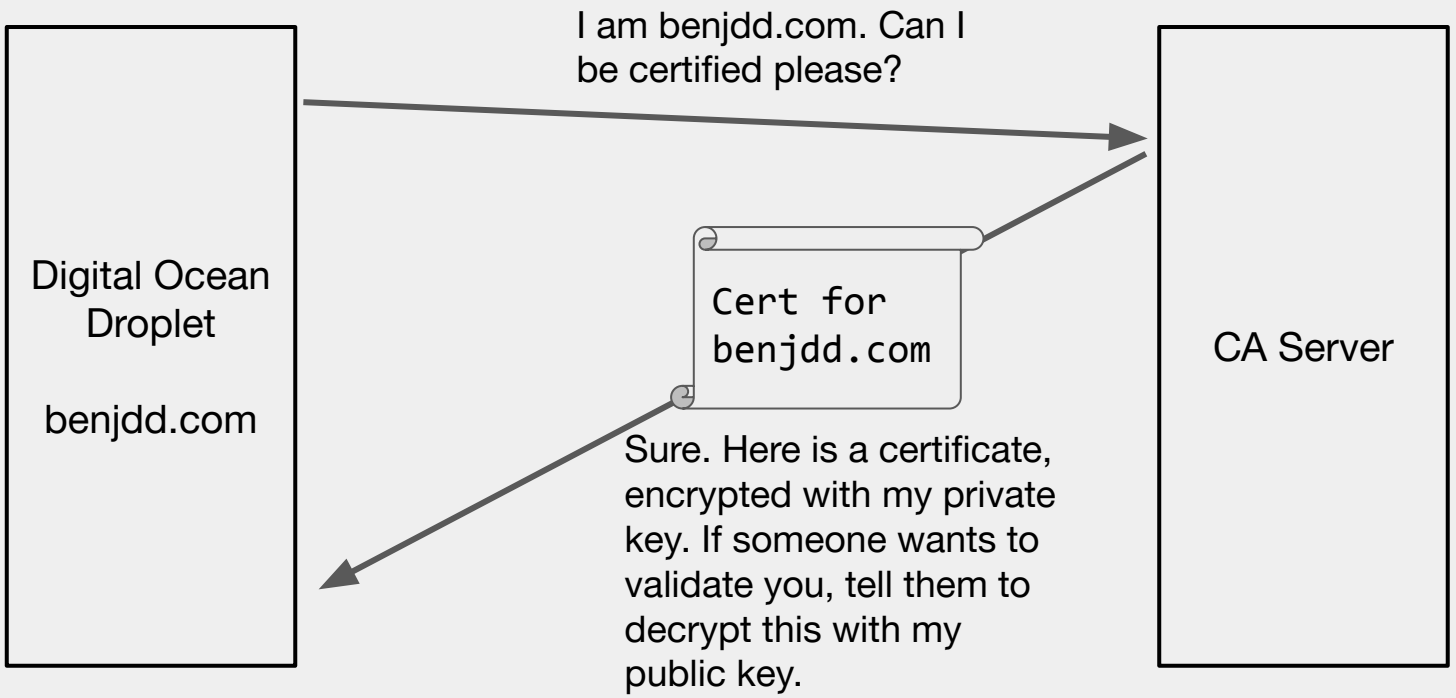
HTTPS

- HTTPS is the secure version of HTTP
- HTTP uses plaintext, HTTPS uses public-key crypto
- Think about the “people” as being clients and servers instead

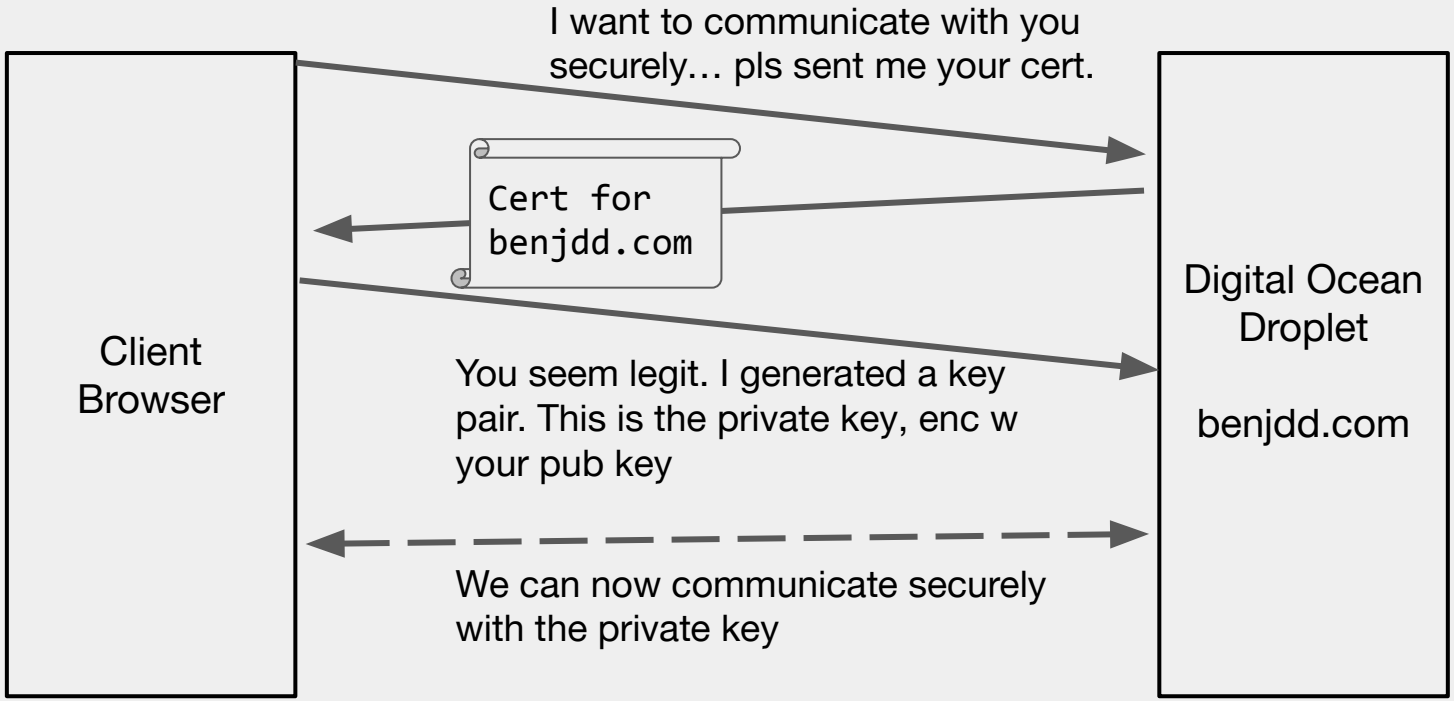
Encryption, Trust, and Certificate Authorities

- Using encryption is great, but how can we tell if we *TRUST* that a website / webserver is who it says it is?
- Certificate authorities can help.

Getting a Certificate from an Authority



Client Validating Authority



Setup HTTPS with Node + Express + LetsEncrypt

Certificate Authorities

- Popular CAs include: IdenTrust, DigiCert, Sectigo, Lets Encrypt
- ***LetsEncrypt*** is a nonprofit certificate authority that provides free certificates
- **Certbot** is a tool that gives you the ability to create certificates via Lets Encrypt