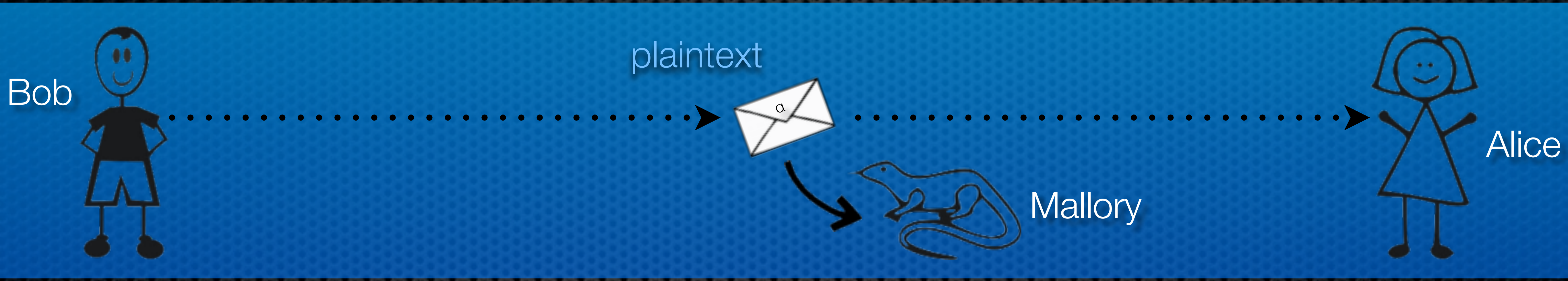
Information Security

Protecting your data in the new age

presented by David Tucker

People are NOT always honest.

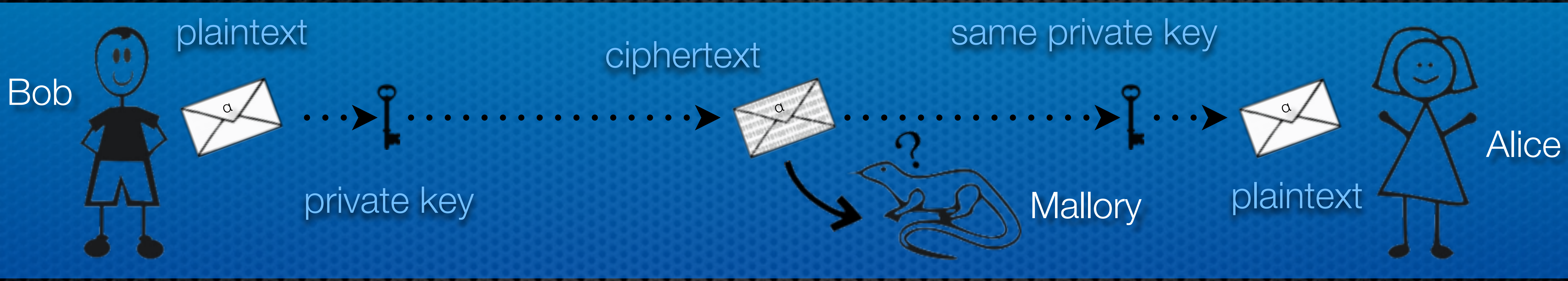
Scenario



Attack	Eavesdropping (a.k.a. Packet Sniffing)
Defense	Encryption
Note	There are 2 types of encryption.

Treat ALL people as untrusted.

Scenario

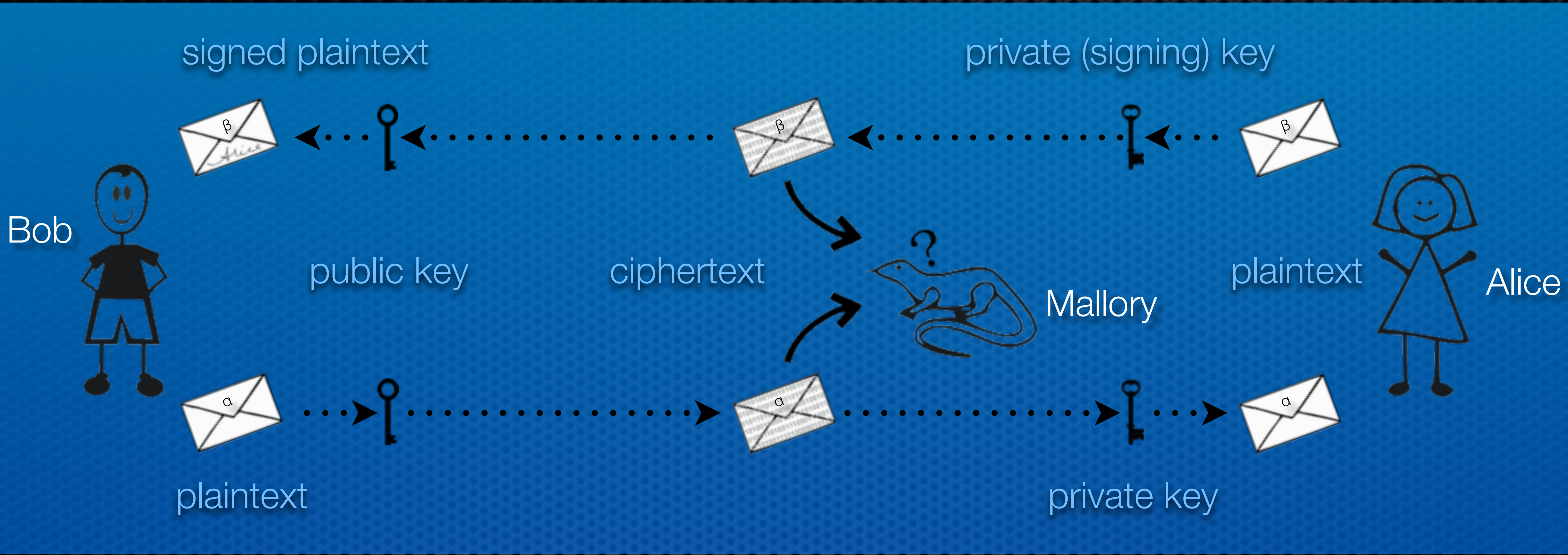


Attack — Eavesdropping (a.k.a. Packet Sniffing)

Defense — Symmetric Encryption

Note — The shared private key must be kept secret.

Scenario



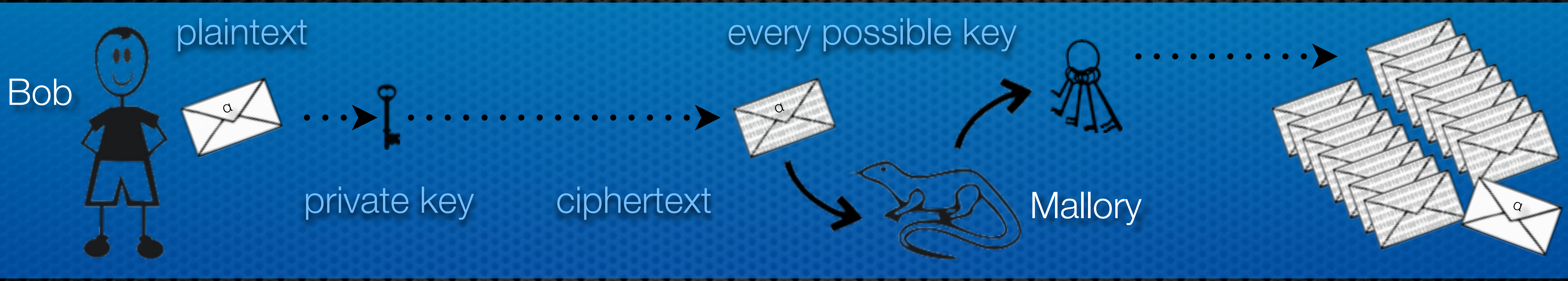
Attack Eavesdropping (a.k.a. Packet Sniffing)

Defense Asymmetric Encryption

Note This is used in TLS (HTTPS) and SSL (deprecated).
There are other ways to digitally sign a message.

Guessing is always an option.

Scenario



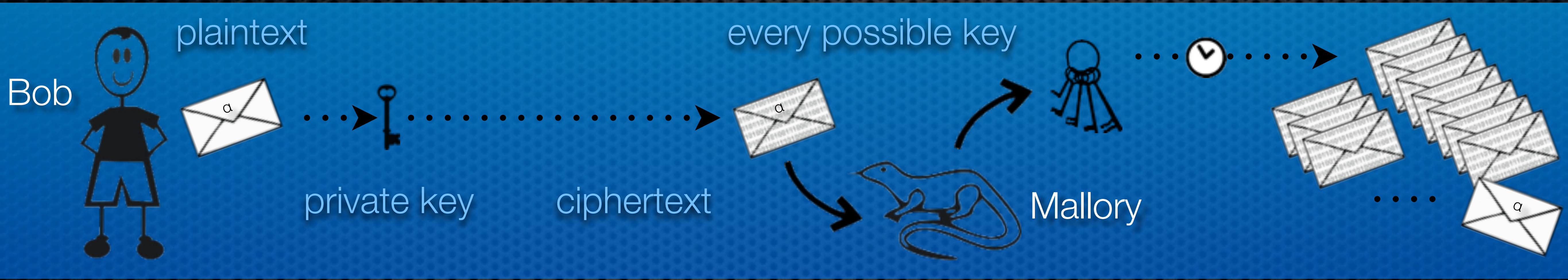
Attack Brute Force (a.k.a. Exhaustive Key Search)

Defense Long Keys, Key Stretching

Note Time and computational power are the only obstacles to this attack.

Require time for authentication.

Scenario



Attack Brute Force (a.k.a. Exhaustive Key Search)

Defense Key Stretching

Note Alice barely notices a time difference when her key is stretched since she only needs to try 1 key.

Strong attacks are precise.

Example: Ocean's Eleven

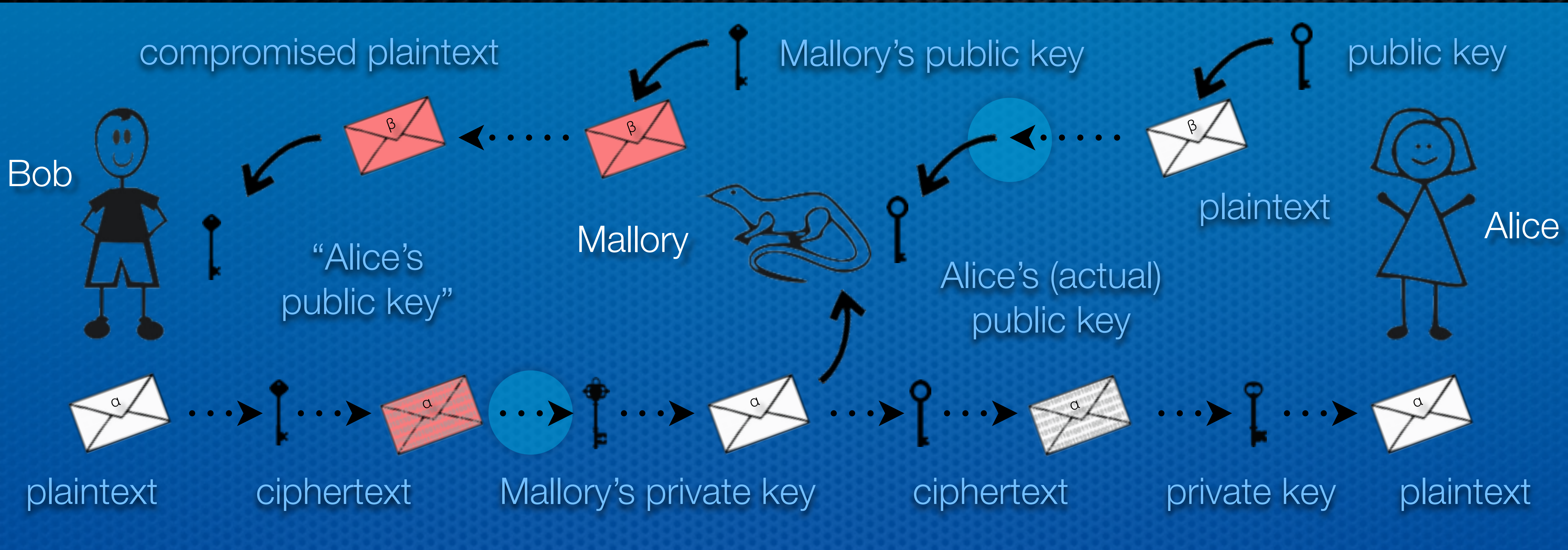
Alice

Bob

Mallory

A good attacker will hide the attack.

Scenario

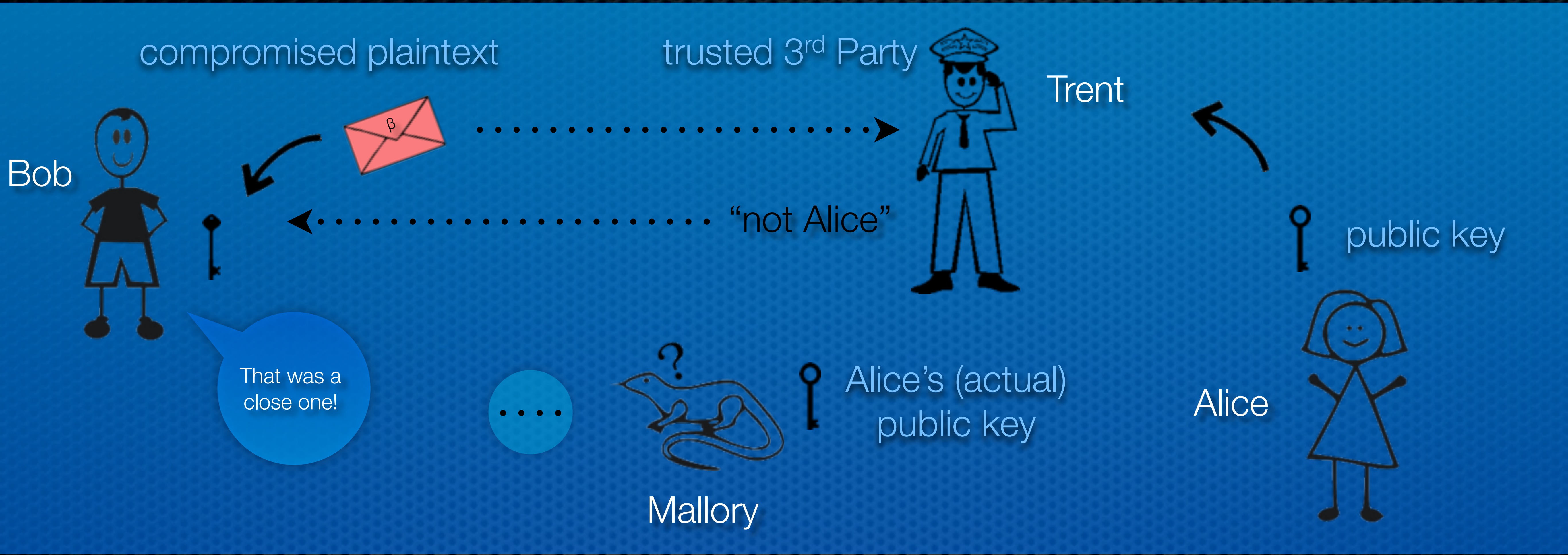


Attack Man in the Middle (MiM)

Defense 2-Factor Authentication
Digital Signatures

Note Mallory must intercept at the circled regions.

Scenario



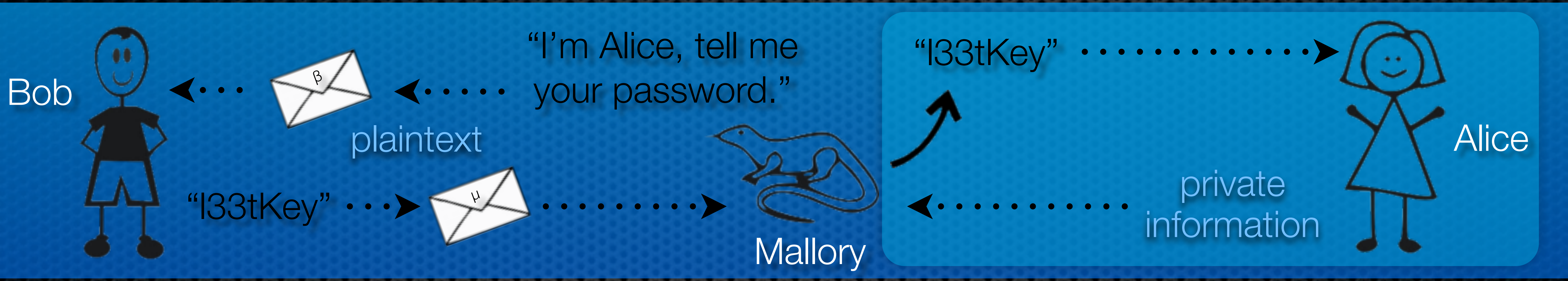
Attack — Man in the Middle (MiM)

Defense — 2-Factor Authentication
Digital Signatures

Note — This is why good SSL certificates cost money.

Solicitous MiM traps are everywhere.

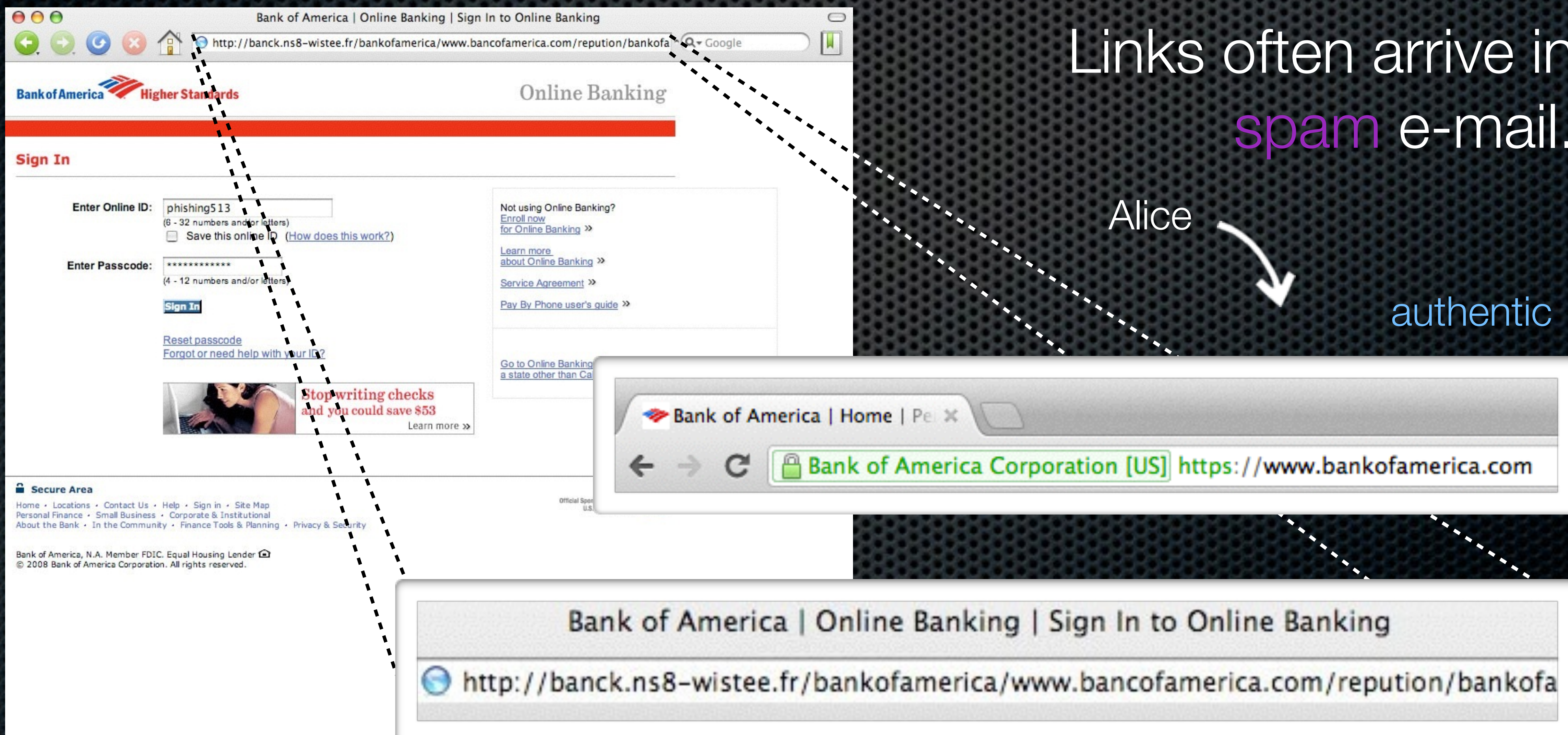
Scenario



Attack	Phishing
Defense	Careful Attention, Digital Signatures
Note	This is a prominent method of identity theft. Transmissions in the rectangle may be encrypted.

Phishing attacks are exact replicas.

Example: Bank of America



Links often arrive in spam e-mail.

Alice

authentic

faked

Mallory

Assuming you are not deceived into providing your key...

The easiest way to provide maximal security is to use strong keys.

That means...

- ☛ hard to predict (random)
- ☛ among many possibilities (long).

Note Random is NOT the same as psuedo-random.

One-time Pads are unbreakable.

"Breaking a cipher simply means finding a weakness in the cipher that can be exploited with a complexity less than brute force."

~Bruce Schneier

That means...

- truly random (observed)
- as long as the message
- completely secret and only used once

Since OTPs are not always feasible...

We must find a balance between secure and memorable keys.

That means...

- ☛ Do NOT use dictionary words.
- ☛ Use multiple character sets (e.g. a, A, 1, !).
- ☛ Use passphrases instead of passwords.
- ☛ Change keys often.

the (arguably) best way to generate memorable keys

Ingredients

- true random number generator (https://www.random.org/integers/)
- well-populated word list or dictionary (http://wordlist.sourceforge.net/)
- a text editor with line numbers

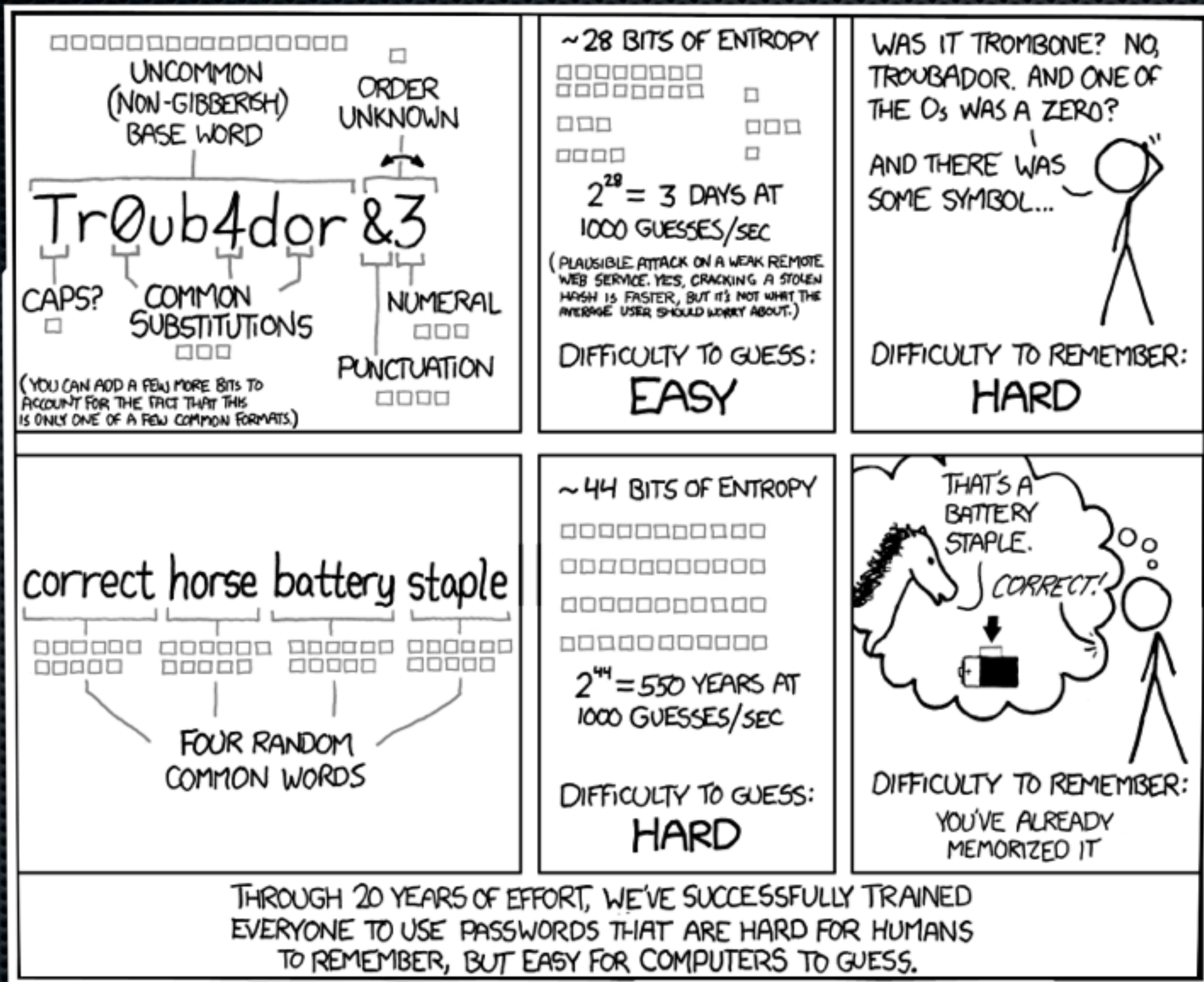
Algorithm

1. Set the maximum number of the generator to the number of words in your list or dictionary.
2. Make sure the generator is NOT checking for uniqueness of generated numbers.
3. Generate ≥ 3 random numbers (depending on how often the key will be changed).
4. Open the word list or dictionary in the text editor.
5. Go to the line indicated by each random number and concatenate each corresponding word to create a passphrase.

Example

correct horse battery staple

XKCD



UNCOMMON (NON-GIBBERISH) BASE WORD
ORDER UNKNOWN
Tr0ub4dor &3
CAPS?
COMMON SUBSTITUTIONS
NUMERAL
PUNCTUATION
(YOU CAN ADD A FEW MORE BITS TO ACCOUNT FOR THE FACT THAT THIS IS ONLY ONE OF A FEW COMMON FORMATS.)
~28 BITS OF ENTROPY
2^28 = 3 DAYS AT 1000 GUESSES/SEC
(PLAUSIBLE ATTACK ON A WEAK REMOTE WEB SERVICE. YES, CRACKING A STOLEN HASH IS FASTER, BUT IT'S NOT WHAT THE AVERAGE USER SHOULD WORRY ABOUT.)
DIFFICULTY TO GUESS: EASY
WAS IT TROMBONE? NO, TROUBADOR. AND ONE OF THE Os WAS A ZERO?
AND THERE WAS SOME SYMBOL...
DIFFICULTY TO REMEMBER: HARD
correct horse battery staple
FOUR RANDOM COMMON WORDS
~44 BITS OF ENTROPY
2^44 = 550 YEARS AT 1000 GUESSES/SEC
DIFFICULTY TO GUESS: HARD
THAT'S A BATTERY STAPLE.
CORRECT!
DIFFICULTY TO REMEMBER: YOU'VE ALREADY MEMORIZED IT
THROUGH 20 YEARS OF EFFORT, WE'VE SUCCESSFULLY TRAINED EVERYONE TO USE PASSWORDS THAT ARE HARD FOR HUMANS TO REMEMBER, BUT EASY FOR COMPUTERS TO GUESS.

Why is this way best?

Say Bob uses a 10,000 word dictionary and a true random number generator to derive a passphrase of 4 concatenated words. Let's also assume Mallory (who knows how Bob's key was created) wants to break this key and has the ability to guess 2,000 keys per second.

How long (at most) will it take Mallory to brute force Bob's passphrase?

Odds of a correct guess:

$$\frac{1}{10^4} \cdot \frac{1}{10^4} \cdot \frac{1}{10^4} \cdot \frac{1}{10^4} = \frac{1}{10^{16}}$$

Time to get the correct key:

$$\left(\frac{10^{16} guesses}{1}\right)\left(\frac{1 second}{2,000 guesses}\right) = 5(10^{12}) seconds$$

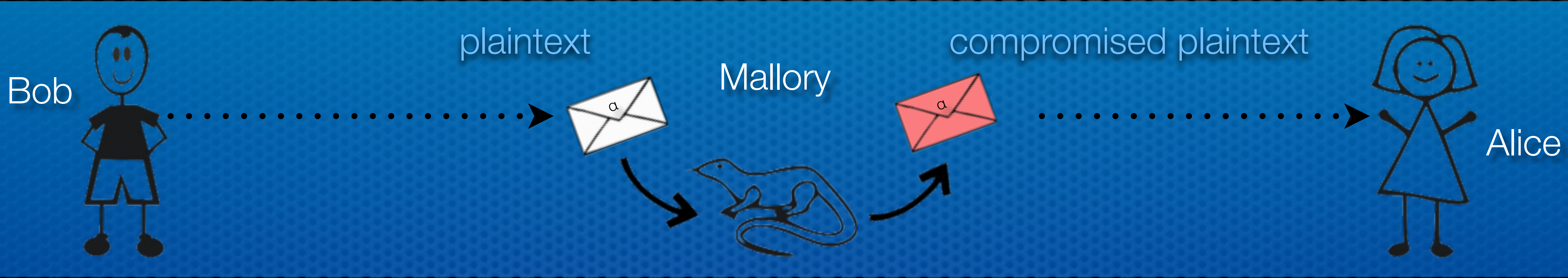
$$\approx 158440 years$$

Bottom line:

Most dictionaries have far more than 10,000 words, and most attackers will not know your word list or even your key generation algorithm!

Encryption is only half of the story.

Scenario



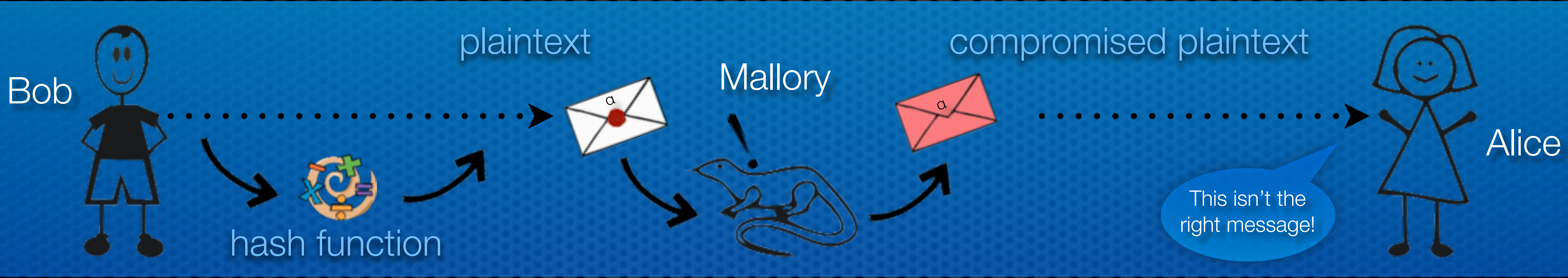
Attack — Tampering (substitution)

Defense — Hashing, Digital Signatures

Note — Tampering with ciphertext is generally less useful.

Hashing maintains integrity.

Scenario



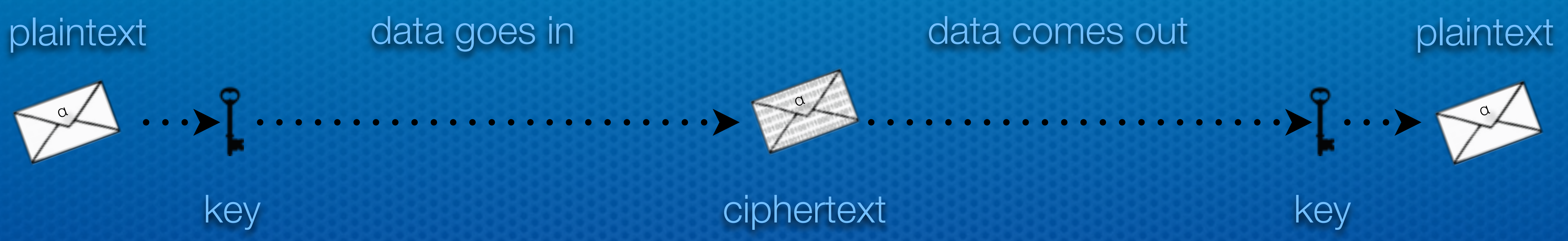
Attack — Tampering (substitution)

Defense — Hashing, Digital Signatures

Note — The seal is (ideally) always unique to the message. Alice will either notice a missing or incorrect seal.

encryption v. hashing

Encryption uses a 2-way function.



Hashing uses a 1-way function.

Since the data remains in the digest...

Hashes are often used for comparison.

Applications include...

- ☛ Message Integrity
- ☛ Digital Signatures
- ☛ Key Storage
- ☛ Authentication

Do NOT store plaintext passwords.

Scenario

Attack — Theft

Defense — Hashing (preventative)

Note — It is actually that easy.

Hashing is vital to key storage.

Scenario

Attack — Theft

Defense — Hashing (preventative)

Note — Trent does NOT know the plaintext keys anymore.

Comparing is faster than hashing.

Scenario

Attack — Rainbow Tables

Defense — Hashing and Salting (preventative)

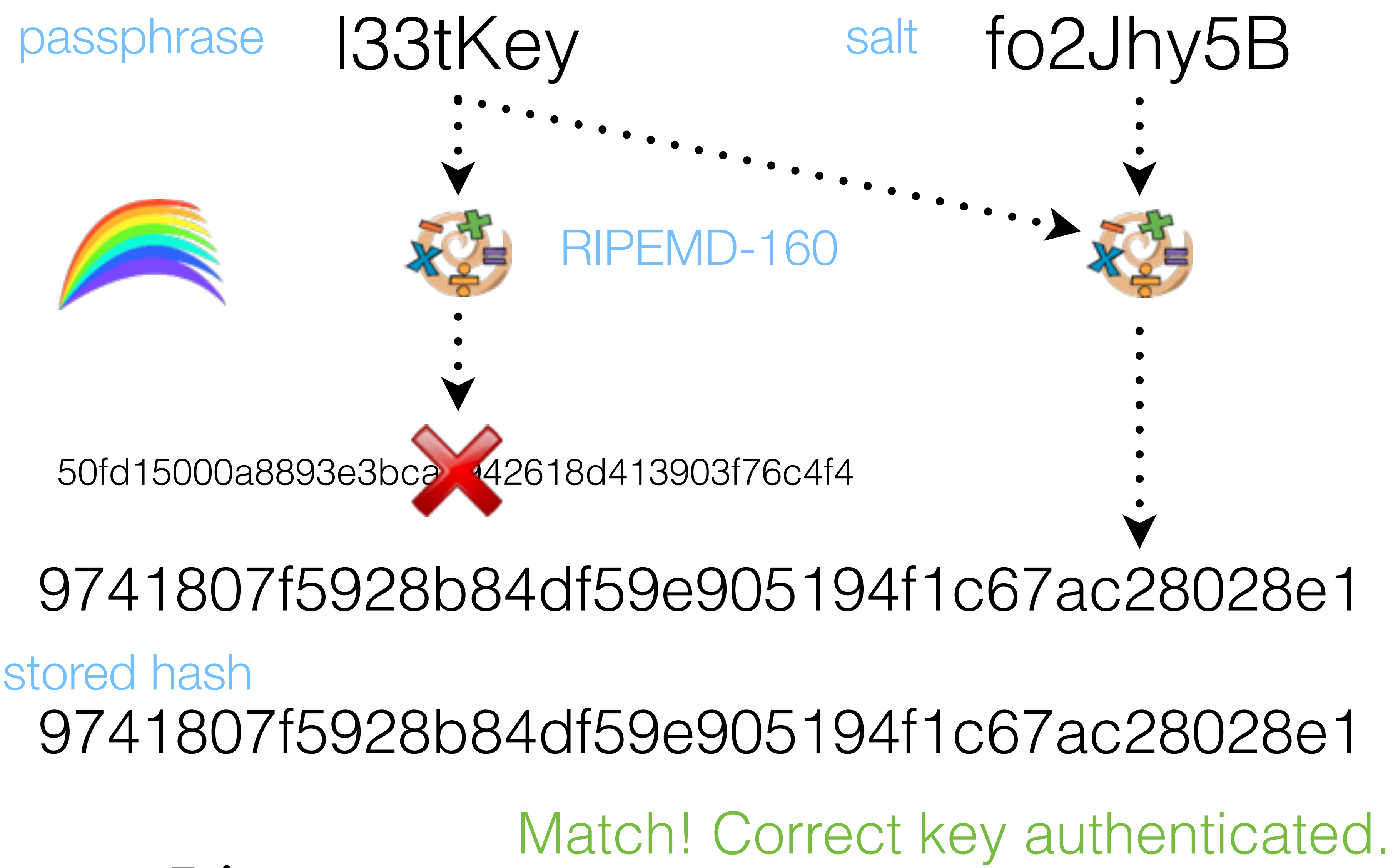
Note — This is a form of brute force.

Salting separates the colors.

Scenario

Attack	Rainbow Tables
Defense	Hashing and Salting (preventative)
Note	Recall the Key Stretching technique.

Example



Bottom Line

An attacker can know the salt!

Knowledge is power.

- *Applied Cryptography* by Bruce Schneier
- http://www.passwordmeter.com/
- http://crackstation.net/hashing-security.htm

Recap.

- 2-way Ciphers (Encryption)
  - Symmetric
  - Asymmetric
    - TLS (HTTPS)
    - Digital Signatures
  - Attacks & Defenses
    - Brute Force & Key Stretching
    - MiM & 2-factor Authentication
    - Phishing & Clues

- Strong Keys
  - One-time Pads
    - Cipher Breakability
  - Password Guidelines
  - Secure Passphrases
    - Generation
    - Analysis

- Protocols
  - Key Exchange
  - Time Stamping
  - Authentication
  - Attacks & Defenses

- 1-way Ciphers (Hashing)
  - Applications
    - Message Integrity
    - Key Storage
    - Digital Signatures
  - Attacks & Defenses
    - Rainbow Tables & Salting

- More
  - *Applied Cryptography* Bruce Schneier