Increasing Security

Reese Hatfield

Encryption

- It’s probably a bad idea to store passwords as plaintext
- When you reset a password
  - Company doesn’t just send you it

- How can they do this?
- Especially knowing hacks happen all the time

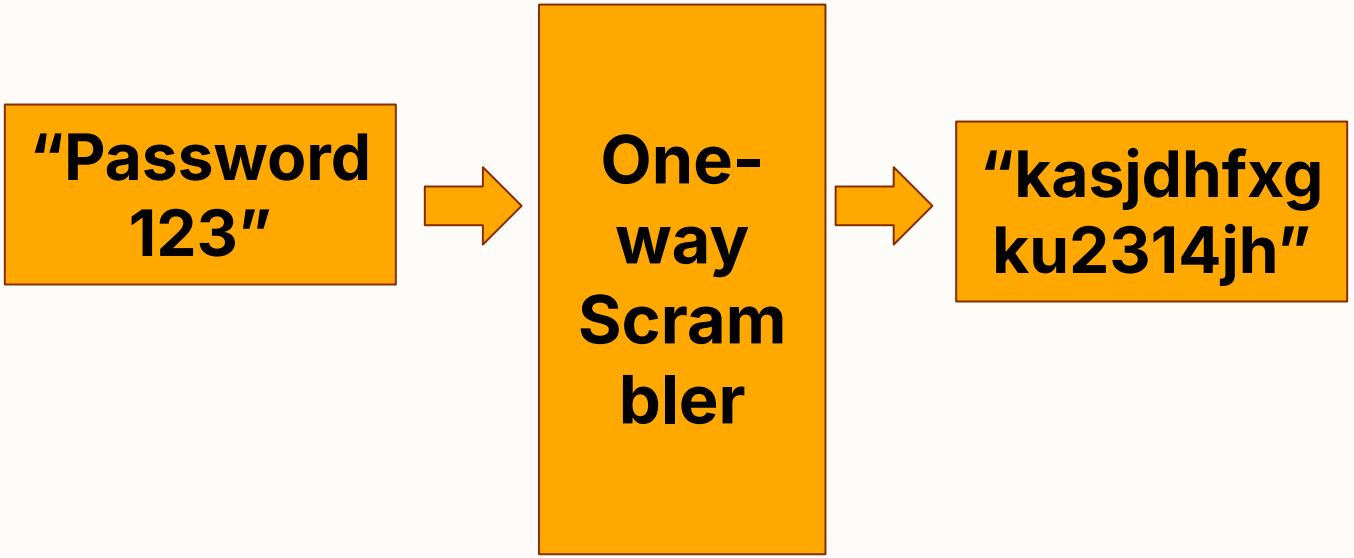
Encryption

- Done via “one-way encryption”
- Turn my password into something that CANNOT be un-encrypted*

- As long as we encryption it the same way

Encryption

- Store only the scrambled data

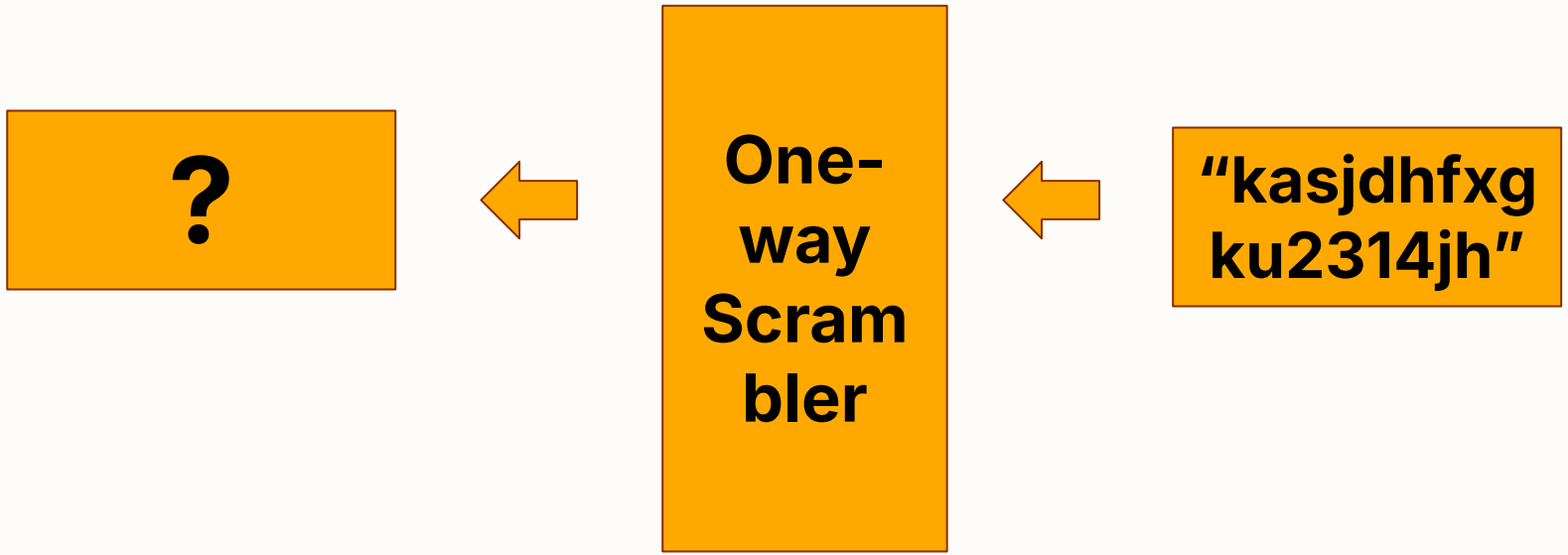


Encryption

- “One-way-encryptors” are called **Hash Functions**
  - Produce hashes/digest
- They are *truly* one way
- If you can go backwards, they are not hash functions

Encryption

- Hash functions cannot be reversible



Encryption

- Common Hash Function:
  - MD5
  - SHA-1
  - **SHA-256**
- We may give you a really simple (and insecure) one to use in your lab

Encryption

- Hash functions must also be deterministic to be useful
- On account creation
  - Create a new password hash
  - Store it
  - On login, check if hashes match
- Sometimes, additional steps added
  - “Salting”

Encryption

- Hashes are still vulnerable to brute force attacks
- If we know the hash algorithm
- We can just try common passwords

- “Rainbow tables”

Encryption

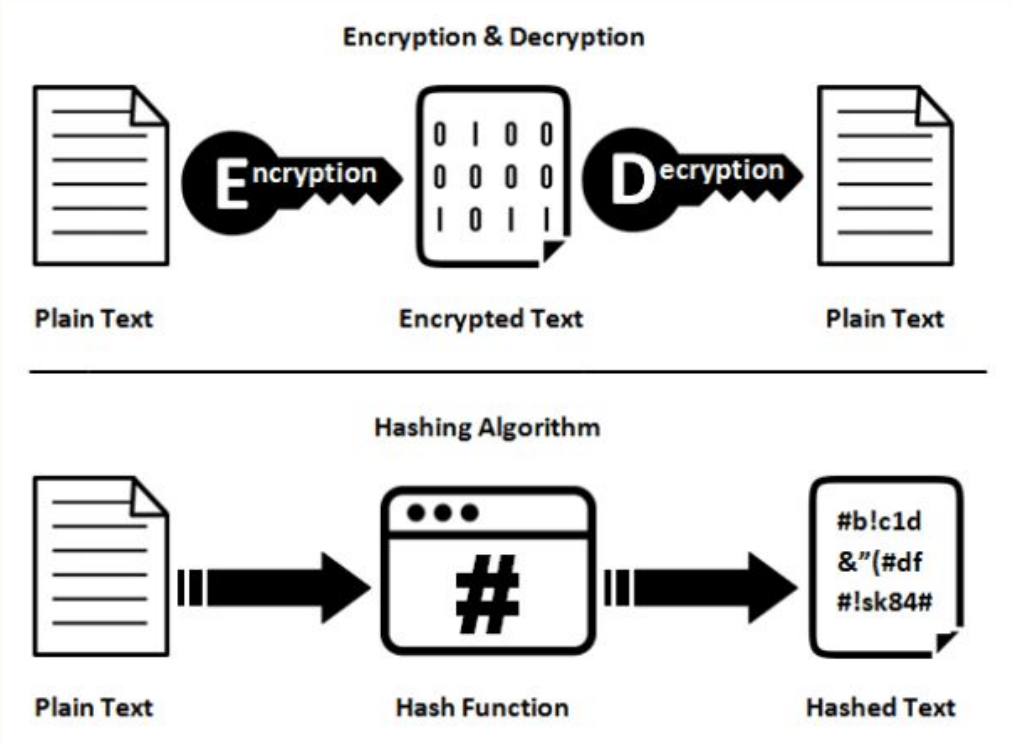
- rockyou.txt
- RockYou ⇒ company who made MySpace and Facebook plugins
- They stored users password
- Did NOT hash
- Data Breach
- 32 Million plaintext passwords

Encryption

- A large amount of people use the same password they did back in 2009
- An attacker could just try every password in rockyou.txt
- Even with hashes
  - Already have the plaintext
- Its already over

Encryption

- Different cryptographic techniques
- Have different uses
- Used in tandem

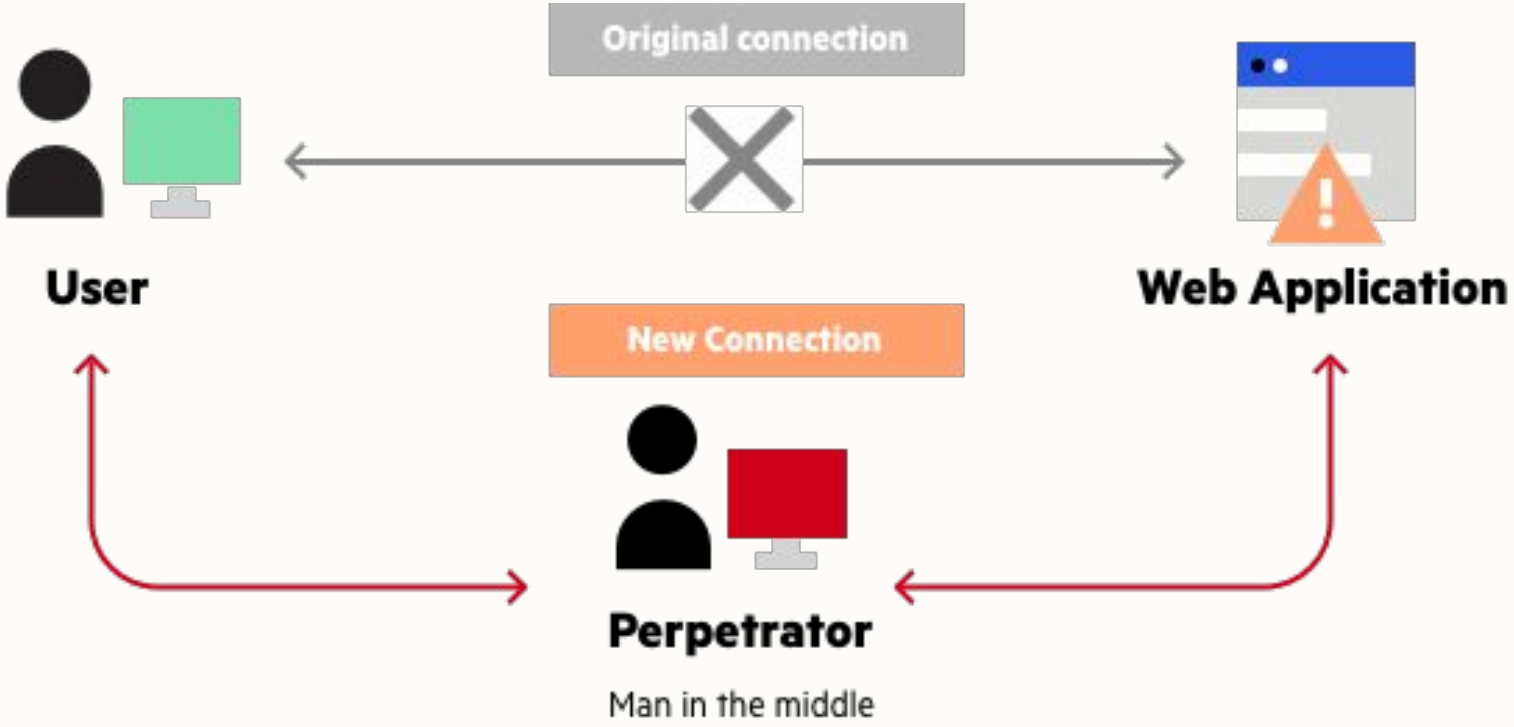


Encryption

- Keep insecure data around for as little time as possible
- OTA ⇒ “Over the air”

- Hash data BEFORE it is sent anywhere
- Prevent people from intercepting vulnerable traffic

“Man in the Middle” Attack



Encryption

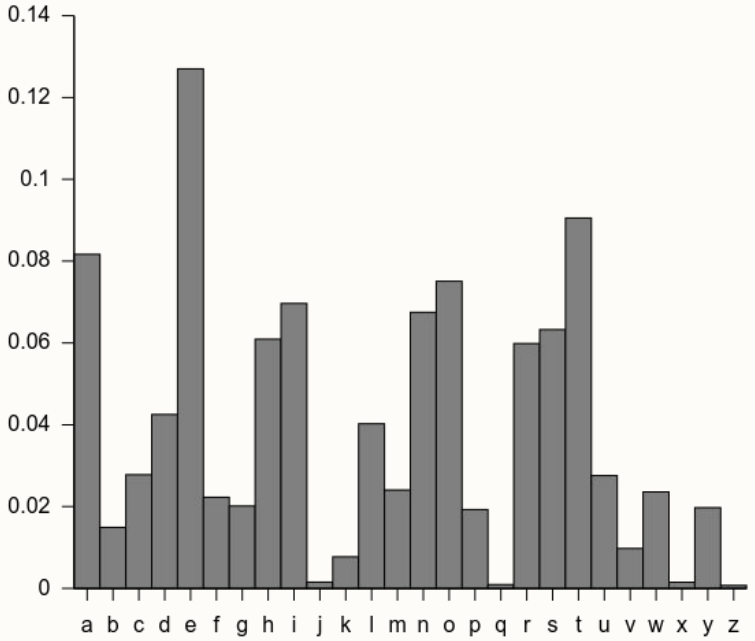
- We said before
  - These encryption techniques are bad
- Caesar ⇒ Try all 26 combination
- Route ⇒ Try a bunch of diff. Spirals
- Hashes ⇒ Try common combinations

Encryption

- This method of “cracking” our codes
- Called “Brute force” attack
- Try every possible combination
  - Goal: make this more difficult
  - How can we do this?
- Increase the time it would take
  - Increase the number of possible combinations

Encryption

- Even if we used a more complex substitution cipher
- Frequency analysis can crack it



Encryption

- How can we *actually* secure our data then?

- Every way of encrypting data so far has been “symmetric”
- Both parties share the same key

Encryption

- Two parties have to keep track of a shared key

- But how did they get the key in the first place?
- Assumed trust

Encryption

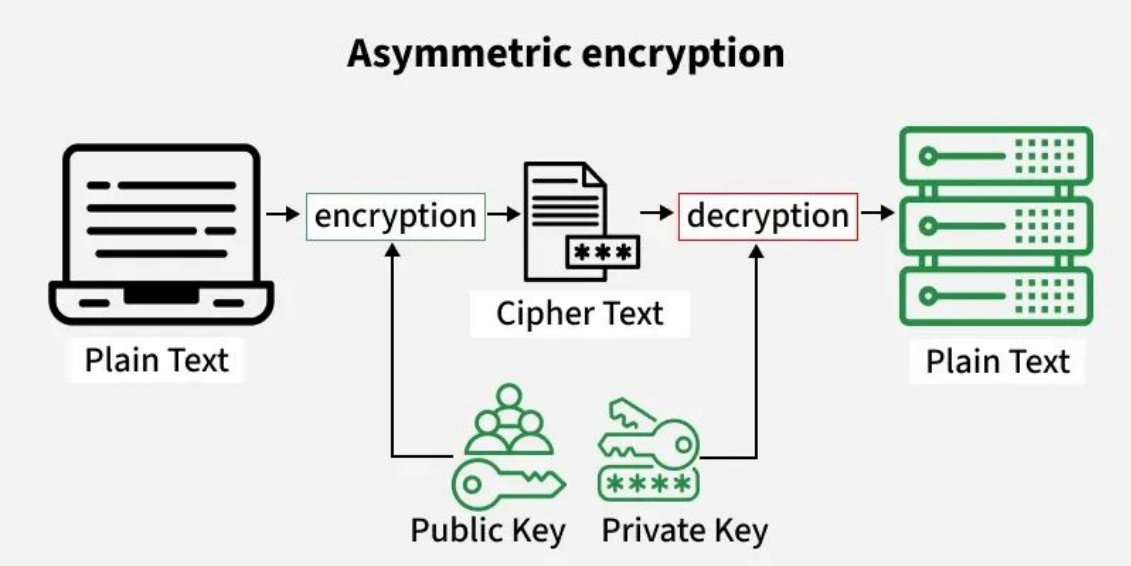
- We really need two separate keys
- Public Key
  - Used to encrypt data
- Private Key
  - Used to decrypt data

Encryption

- If you need to store some users password
- Make a key-pair
  - Hide the private key
  - Let the user see the public key to encrypt the data
- Prevents more “user error”

Encryption

- This is more generally called
- **Asymmetric Encryption**



Encryption

- This is still not perfect
- Doesn’t prevent data breaches
  - But is more the industry standard

- All of these techniques are used together
- To create a stronger system of secure encryption