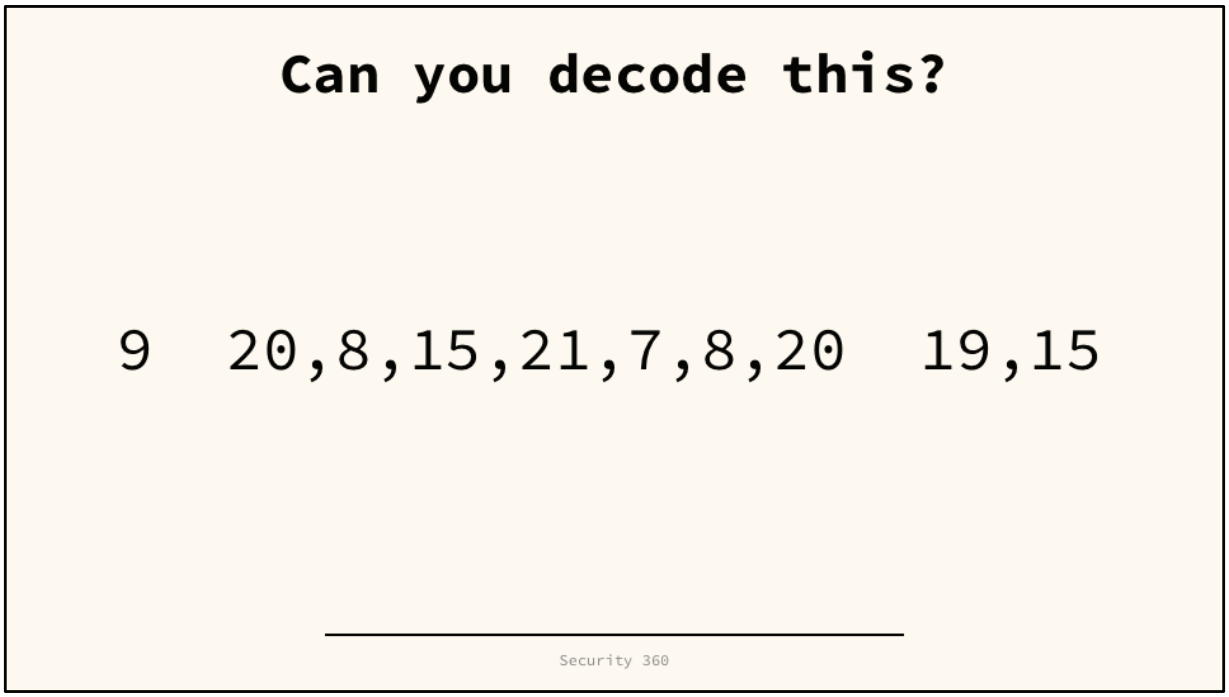
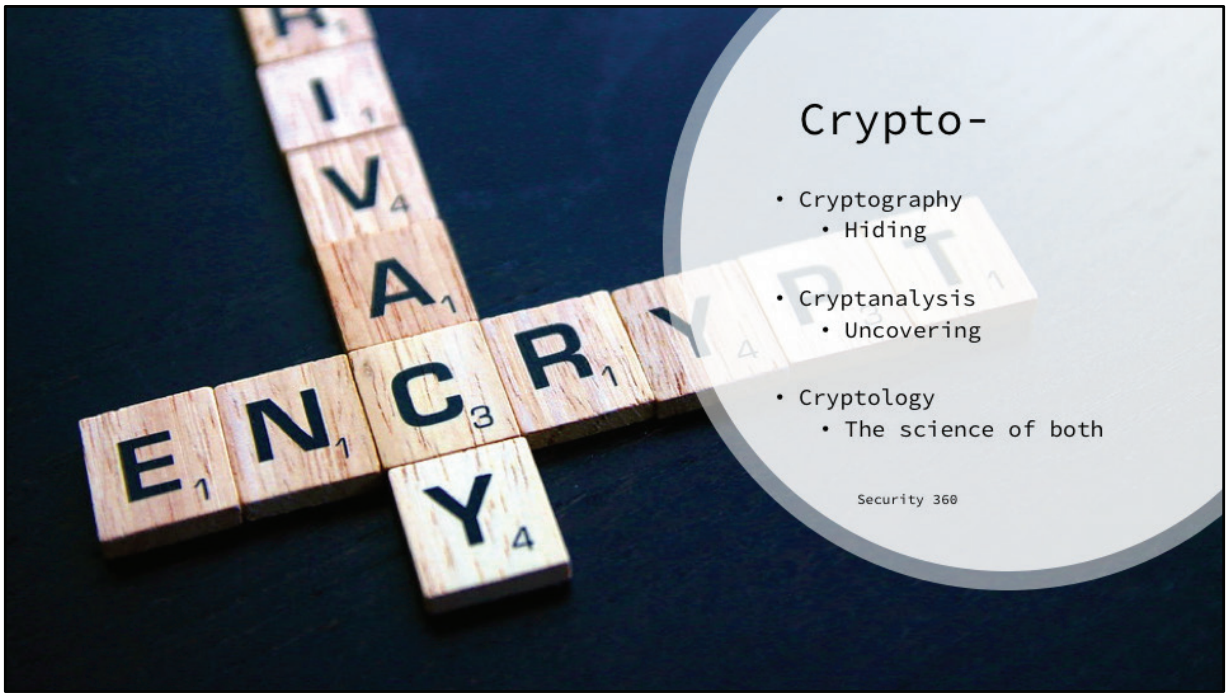
Cryptology

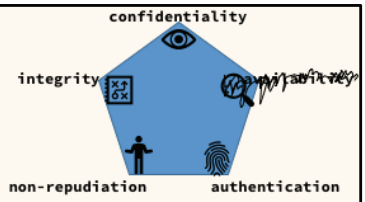
codes and ciphers and keys and more



I thought so



What is it?



- Techniques for secure communication
  - With the help of **ciphers & keys**

principles, means, and methods
for providing information security,
including
confidentiality, data integrity,
non-repudiation, and authenticity

NIST SP 800-21

ingcjen?

An (**encryption**) algorithm applied with a **key** to some input (**plaintext**) to produce some output (**ciphertext**), following the principles of:

- Confusion
  - The output is notably different from the input
- Diffusion
  - One change in the input affects much of the output

Ciphers?

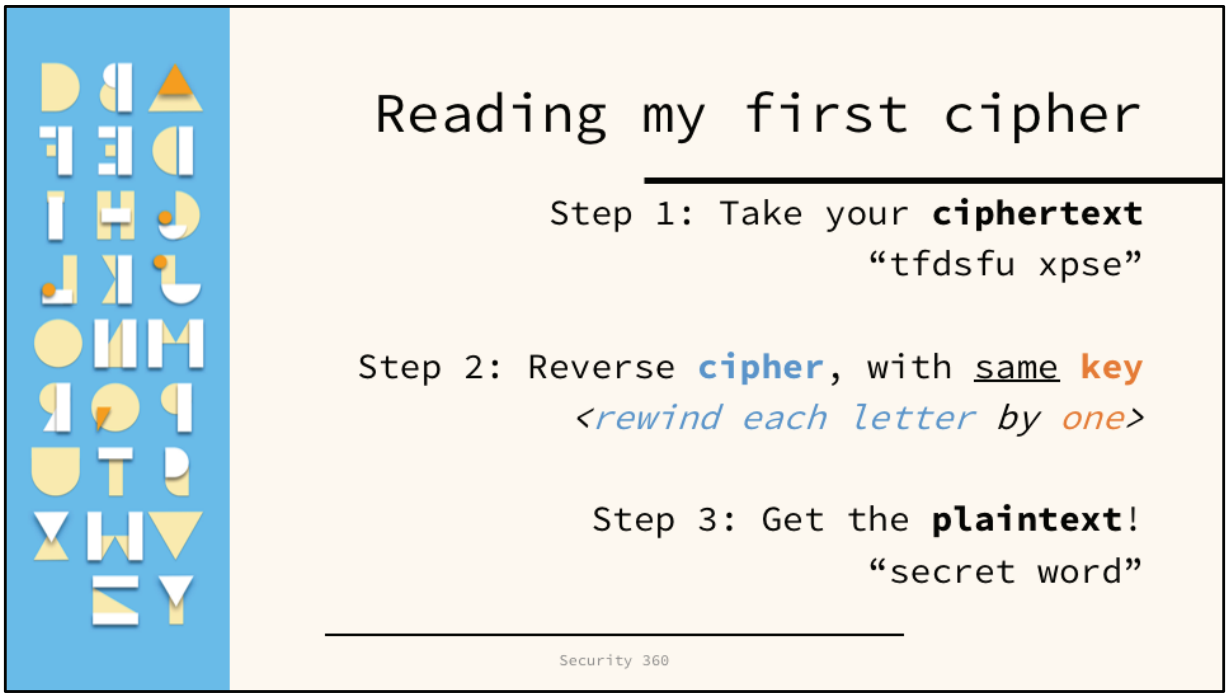
My first cipher

Step 1: Take some **plaintext**
"secret word"

Step 2: Apply a **cipher** & **key**
<*advance each letter by one*>

Step 3: Get your **ciphertext**!
"tfdsfu xpse"

Security 360



Classical Cryptography

- Long has there been a need for securing communication
- The Caesar Cipher
  - One of the simplest
  - But, not the earliest!

Cryptography Through Time

Historical Ciphers (BROKEN)

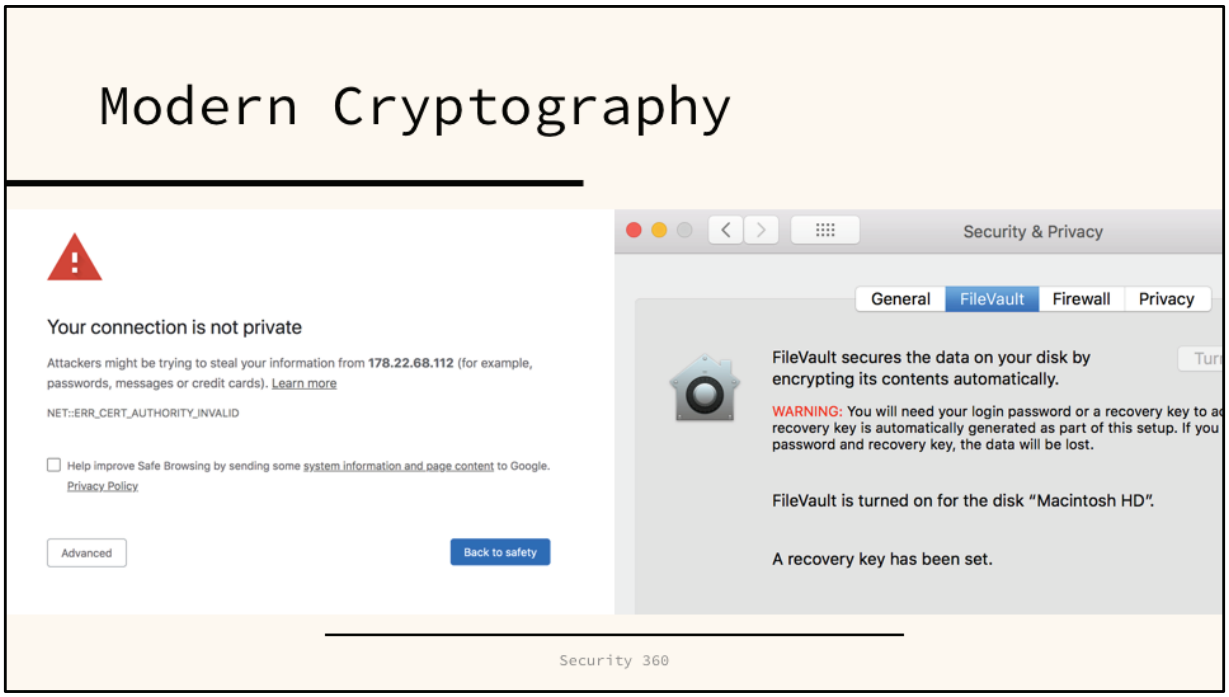
- Substitution Ciphers
  - ROT1 & Caesar (a -> b, b -> c, c -> d)
  - Monoalphabetic (one replacement scheme)
- Vigenere Cipher
  - Multiple Caesars based on a longer, more complex key
  - Polyalphabetic
- Rotor Machines
  - Mechanic polyalphabetic

https://cryptii.com/

Cryptanalysis

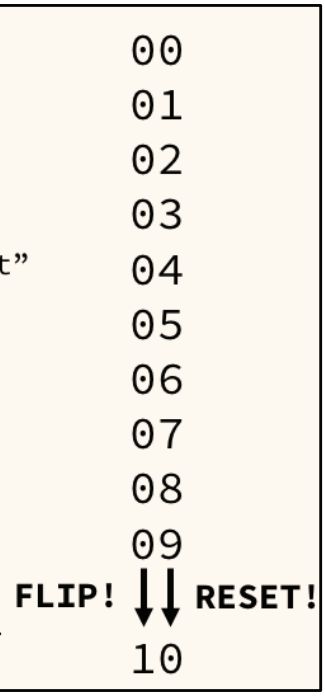
- Getting plaintext from a ciphertext without prior knowledge of the key

- Brute-force
  - Try all the possibilities (Caesar==26)
- Frequency analysis
  - Analysis of common letters of the language



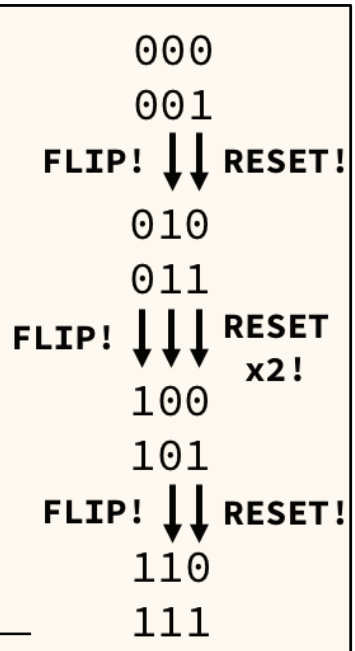
Binary

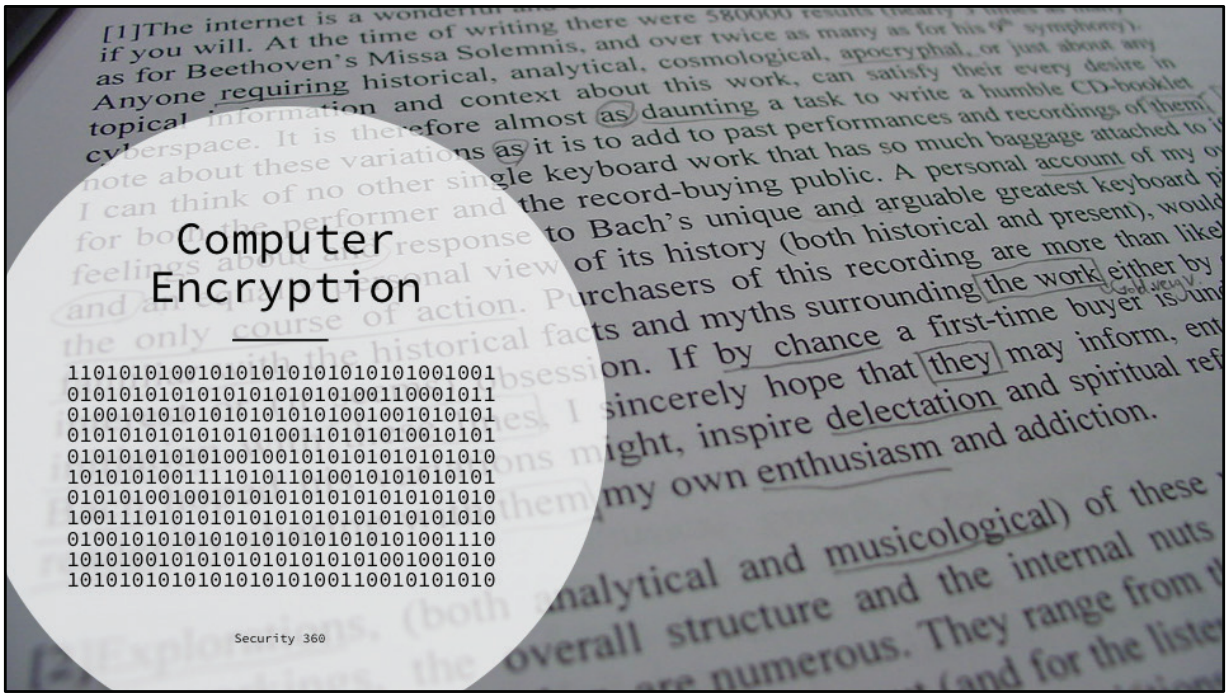
- Human counting is Base**10** (Decimal)
  - We get **10** numbers before we “flip & reset”
  - 0, 1, 2, 3, 4, 5, 6, 7, 8, 9,10,11,12
  - 00,01,02,03,04,05,06,07,08,09,**10**,11,12
  - Leading zeros are always ignored
  - After 10 numbers:
    - The leading digit “flips”
    - The trailing digit(s) “resets”



Base2

- Computer counting is Base2
  - The *same* principles apply, but now we only have 2 numbers (0 & 1)
  - 0, 1, ?
  - 00, 01, ?
  - 00, 01, 10, 11
  - Remember, leading zeros are ignored!





A very simple cipher

- Exclusive or
- XOR $\oplus$
- Either, but not both

$0 \oplus 1 = \text{True (1)}$
$1 \oplus 1 = \text{False (0)}$
$0 \oplus 0 = 0$
$1 \oplus 0 = 1$

Bitwise XOR

Encryption

Plaintext (DEC)	123
Plaintext (BINARY)	1111011
Cipher (XOR)	⊕
Key (BINARY)	0100011
Ciphertext (BINARY)	1011000

Decryption

Ciphertext (BINARY)	1011000
Cipher (XOR)	⊕
Key (BINARY)	0100011
Plaintext (BINARY)	1111011

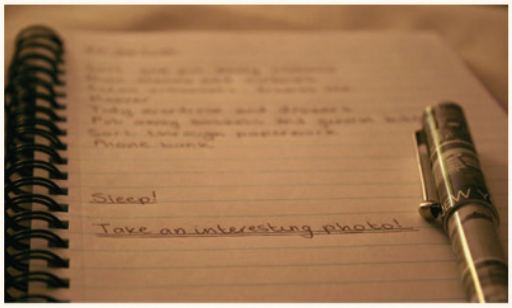
XOR in Practice

- Any key can be cracked with the plaintext
  - ciphertext ⊕ plaintext = key
- Short keys can be cracked with freq. analysis
- But, a random key, as long as the message, where all components are kept secret is ...

UNBREAKABLE (in theory)

Stream Ciphers

- When a pseudo-random number generator streams (: generates on the fly) a key as long as the plaintext and performs bitwise XOR
- The result
  - One-Time Pad (OTP)



Block Ciphers

- Data gets chunked into “blocks” and encrypted
- Bitwise XOR is not as effective here, so it is not used alone
- Standards in Block Ciphers
  - DES (broken)
  - 3DES (too slow)
  - AES (works - the standard today)
    - http://www.moserware.com/2009/09/stick-figure-guide-to-advanced.html

http://www.moserware.com/2009/09/stick-figure-guide-to-advanced.html

Block & Stream

- Both fall into the same category of encryption
- Both address the same aspect of infosec

Symmetric Encryption

- AKA private key crypto
- Sought to ensure confidentiality
- Single key to encode/decode

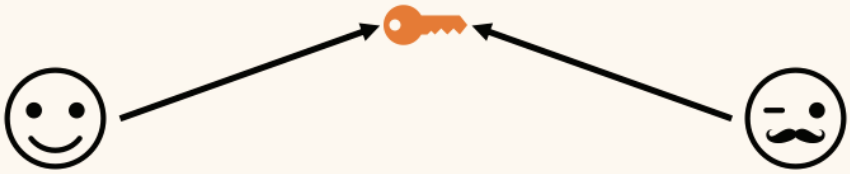
Meet Alice & Bob

Hi I’m Alice

Hi I’m Bob

Key Distribution Problem

- How do Alice & Bob share this key?



Key Exchange Problem

- Can Alice generate the key and send it to Bob?

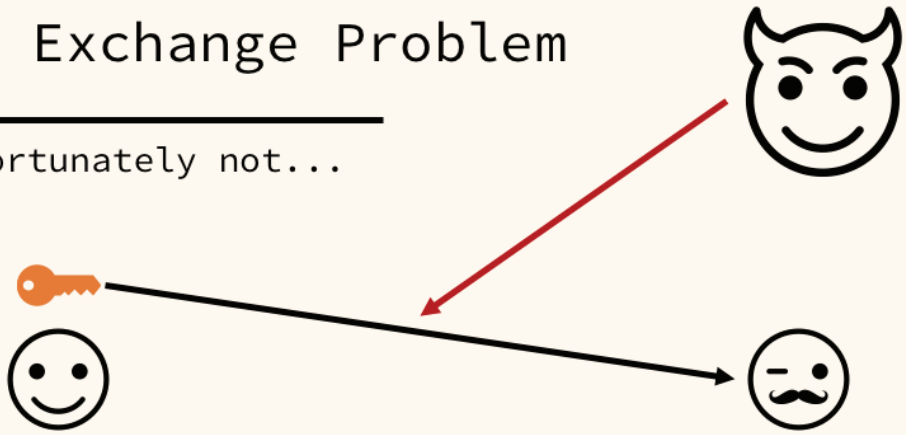
Introducing Eve

Eve

Security 360

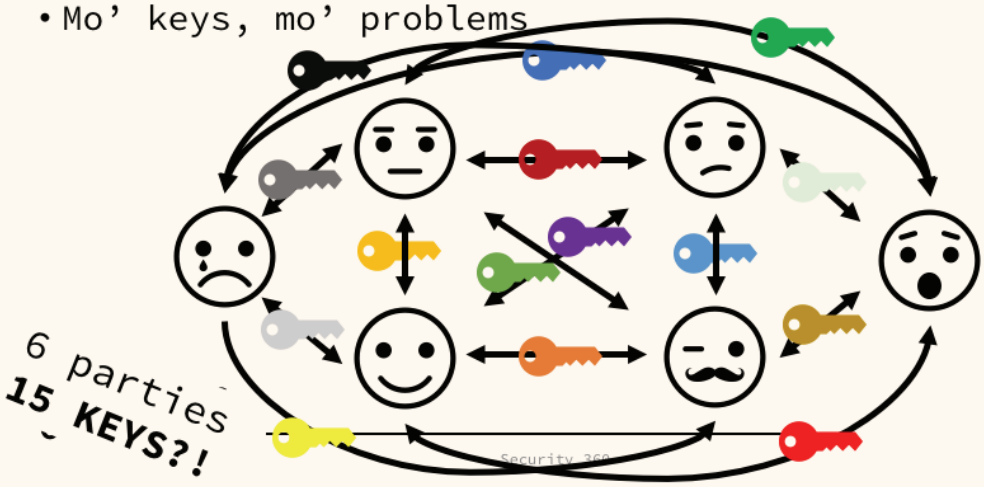
Key Exchange Problem

- Unfortunately not...



More Symmetric Problems

- Mo' keys, mo' problems



Asymmetric Encryption

- Let’s add another key
- Also called public key encryption
- Works on the principle of key pairs
  - Related, but different keys

Key Pairs

- Everyone gets two!
  - A public key, to share with others
  - A private key, to keep secret

Key Pairs

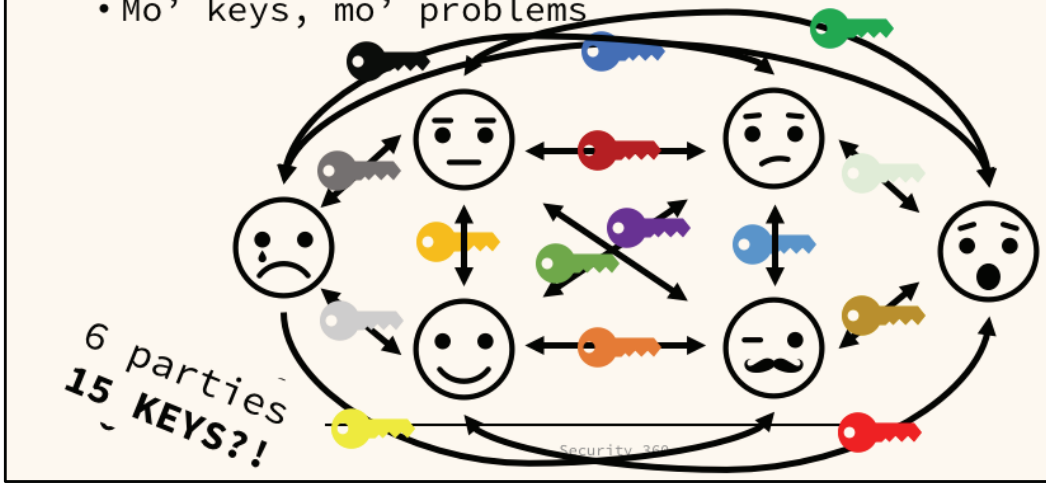
- The relation between the two keys, simplified
  - My public key can lock a box, but not open it
  - Only my matching private key can open that box

Key Pairs

- Others use your public key to encrypt data
- You use your private key to decrypt it!

Remember This?

- Mo' keys, mo' problems

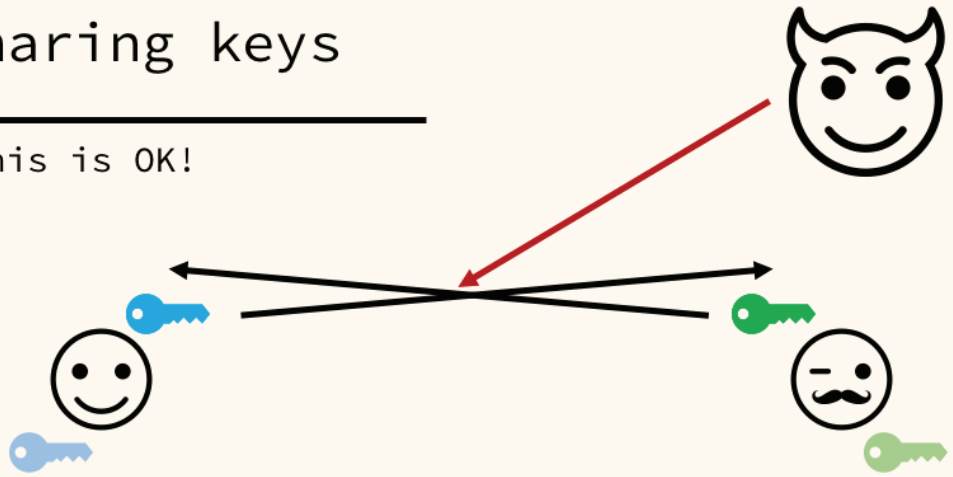


Asymmetric Saves the Day

- Mo' keys, no' problems

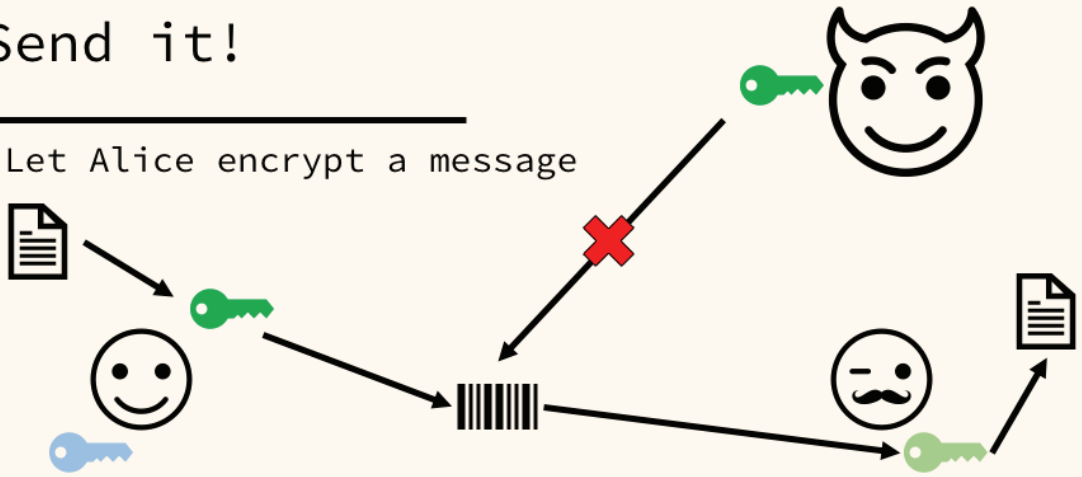
Sharing keys

- This is OK!



Send it!

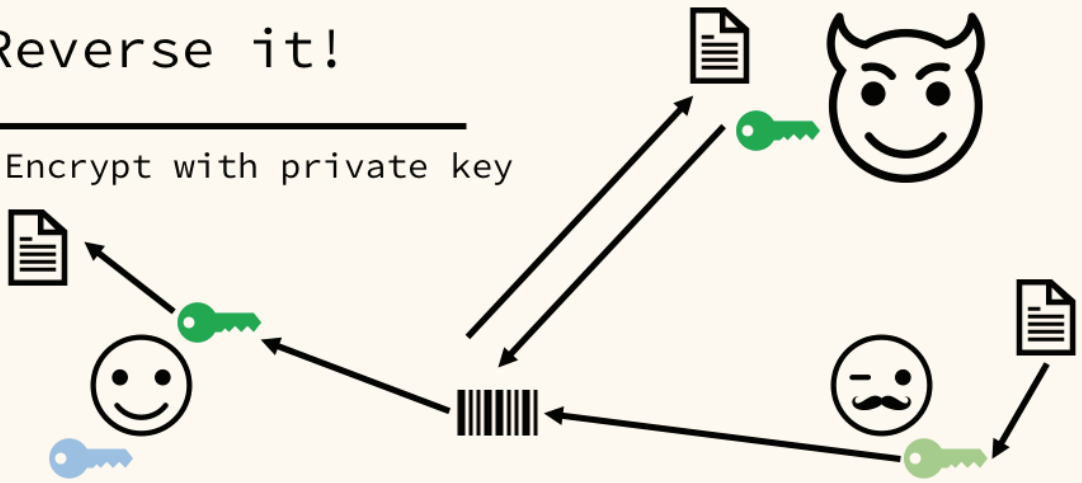
- Let Alice encrypt a message



Let’s try the opposite

Reverse it!

- Encrypt with private key



But... Why would you do that?

- Anyone with the public key (everyone) can decrypt the message
- Effectively, a Digital Signature
- A step to non-repudiation
- Proves the sender, not the content

Asymmetric Isn't Perfect

- Slower (less ideal for confidentiality)
- Still breakable (compromised private key)
- How do we know whose public key that is??

A simple example with SSH

- A common use of asymmetric
  - Authenticating with a server (SSH)

Asymmetric (Exchange) History

- Diffie-Hellman Key Exchange (1976)
  - First to use asymmetric to exchange a symmetric key?
- GCHQ - British Intelligence (1969)
  - First!
- Rivest-Shamir-Adleman (RSA) (1977)
  - An early cryptosystem
  - Takes key exchange, adds **certificates**

Certificates

- A way to prove the owner of the public key
- Works on the basis of a **T**rusted **T**hird **P**arty
  - Certificate Authority

Public Key Infrastructure

- Bob trusts that Certificate Authority too!

In the wild...

- A more familiar scenario

Hi I'm https://ebay.com

Yup that's https://ebay.com

Altogether Now

- A little bit of column A
- A little bit of column B
- A little but of column C

- The best of symmetric with the best of asymmetric, and some certificates

- Speed where it matters

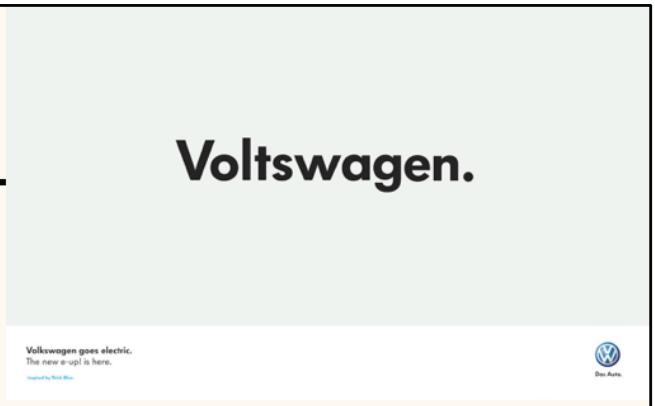
Security 360

Securing Communications Protocols

- Secure Sockets Layer
  - Netscape, 1995
  - Deprecated
- Transport Layer Security
  - Internet Engineering Task Force, 1999
  - Used DH, RSA
  - Now at TLSv1.3
  - Often called SSL/TLS

A Hybrid

- Brings together
  - Key exchange
  - Certificates
  - Asymmetric encryption
  - Symmetric encryption



- DH uses asymmetric encryption to exchange the key later used for symmetric encryption
- Certificates make sure everyone is who they say they are

Hashes

- **Cryptographic** Hashes
  - Another form of encryption
  - No one gets a key, so one-way
  - A.K.A - No decryption possible
- Common uses
  - Storing passwords
  - Ensuring integrity!!

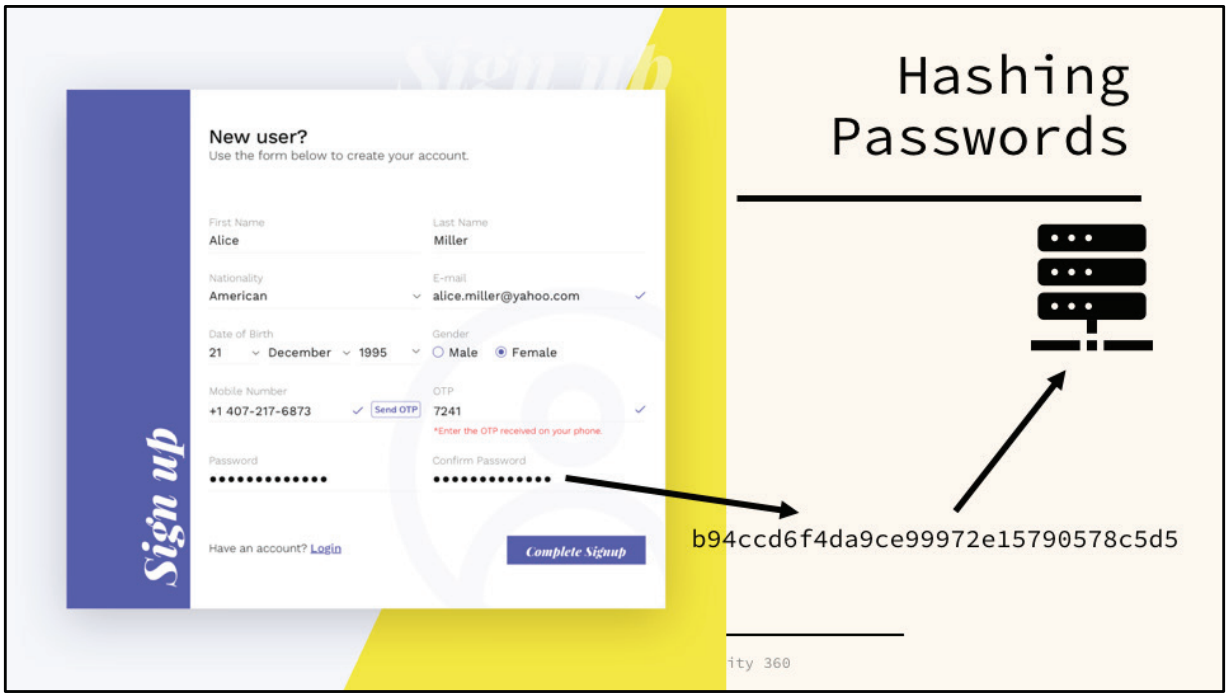
Properties of Hashes

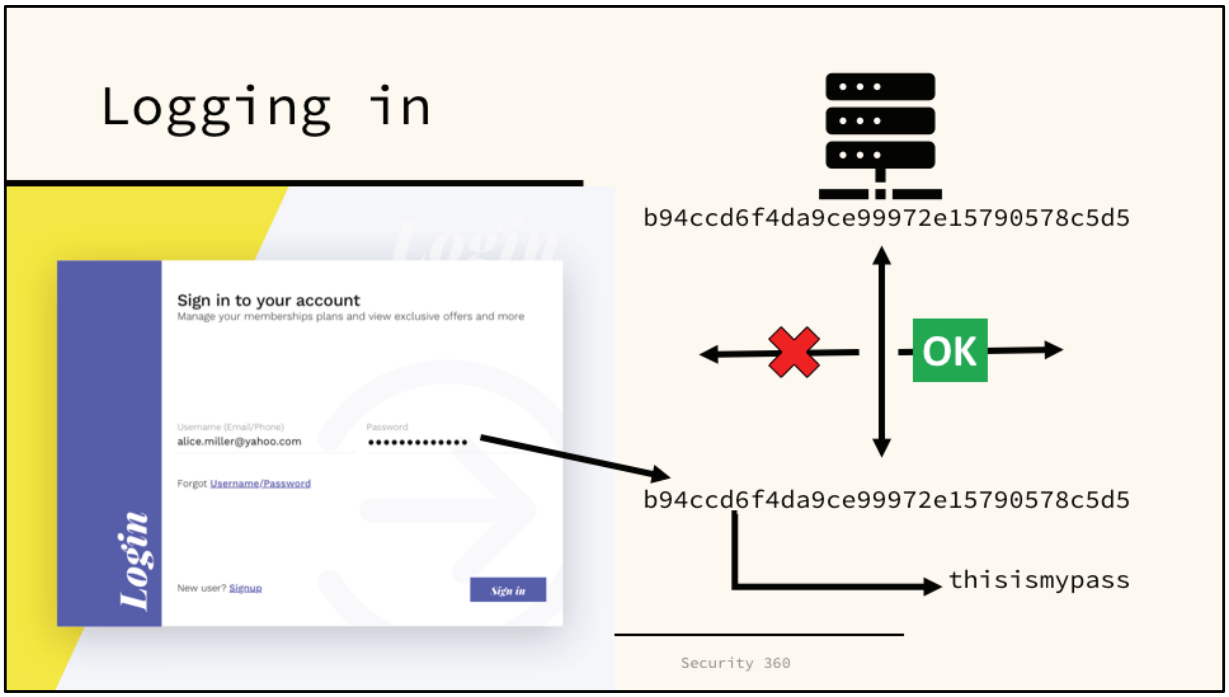
- Easy & quick to calculate
- Extremely difficult to reverse
- Different inputs == unique outputs
  - No two inputs can have the same hash
  - Failure is called a **hash collision**

Some common hash functions

- MD5 – Message Digest 5
  - 128b / 32hex characters
  - Collisions (broken)
- SHA1 (Secure Hash Algorithm)
  - 160b / 40hex
  - Collisions (broken?)
- SHA2 (SHA256 & SHA512)
  - 256b & 512b / 64hex
  - Still good

Security 360





Integrity!

- Hash functions work on **any** number of bits
  - And they're still *pretty* quick!
- Common example of integrity checking
  - Download a file, hash it to ensure its integrity

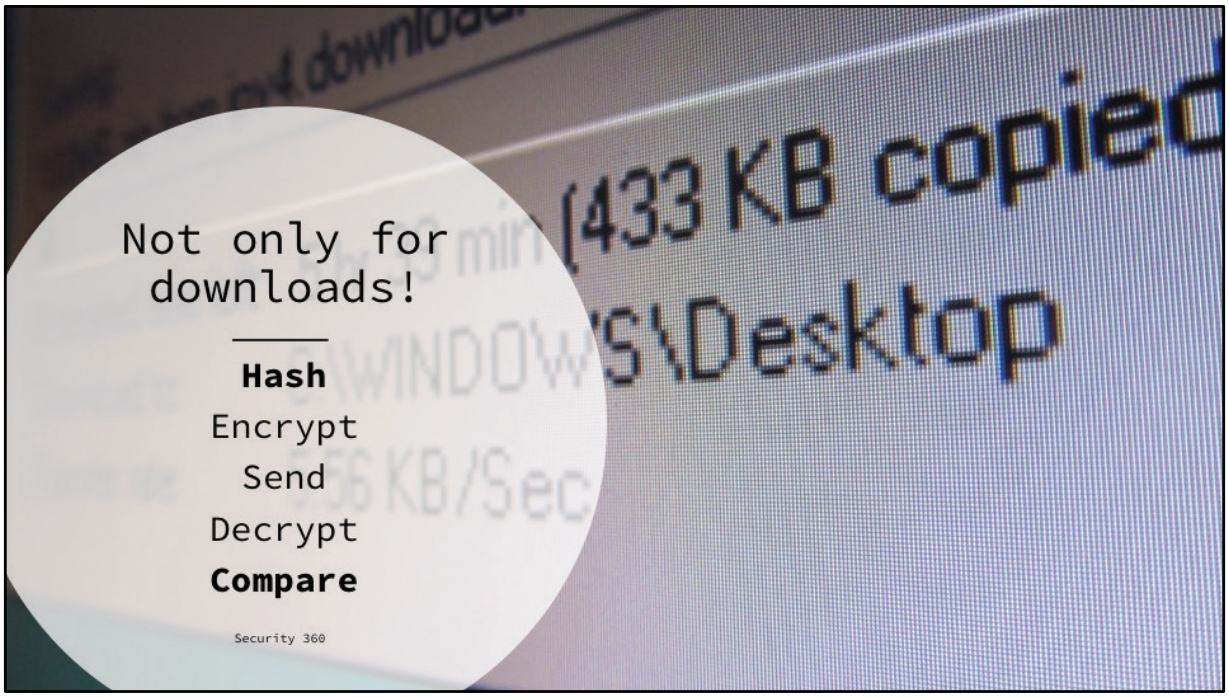
Windows version:

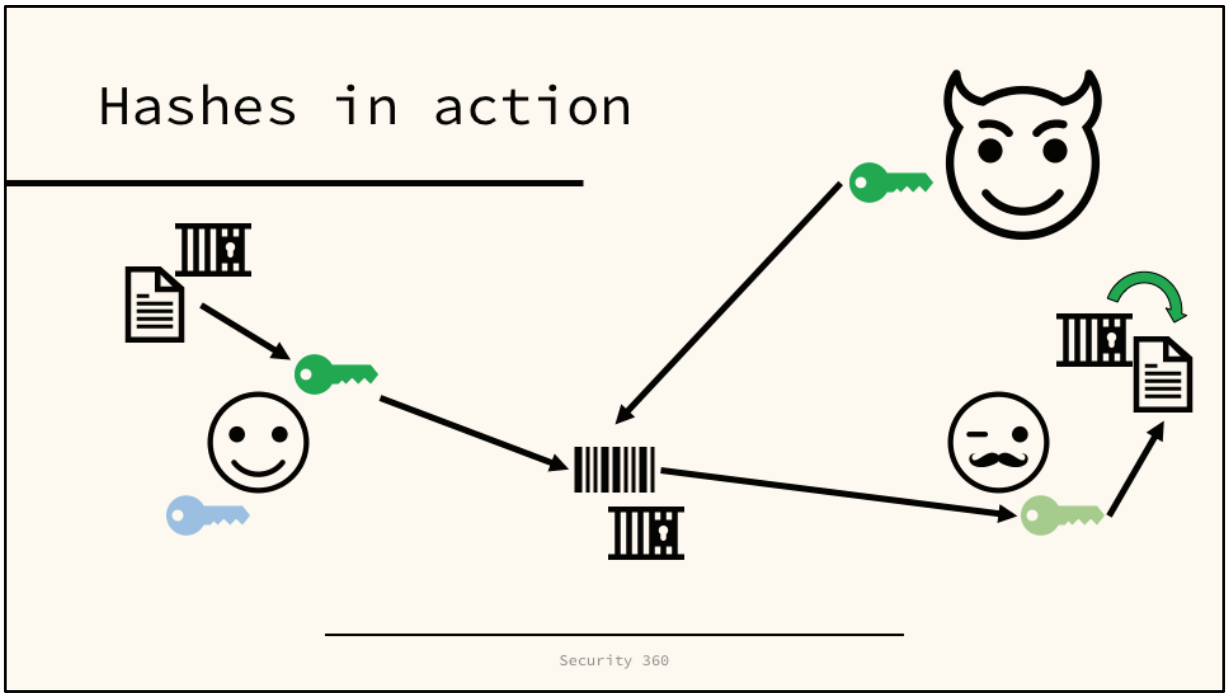
hello.exe. MD5 Sum: cdc47d670159eef60916ca03a9d4a007
erase.exe. MD5 Sum: cdc47d670159eef60916ca03a9d4a007

Linux version (i386):

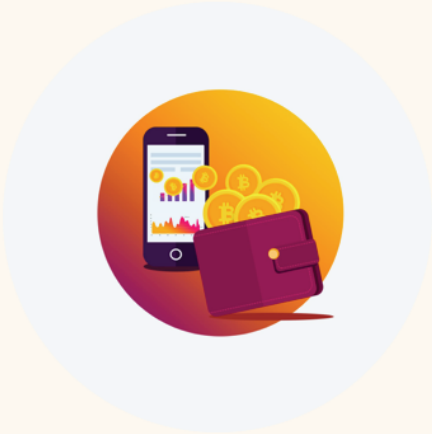
hello. MD5 Sum: da5c61e1edc0f18337e46418e48c1290
erase. MD5 Sum: da5c61e1edc0f18337e46418e48c1290

https://www.mscs.dal.ca/~selinger/md5collision/




Hashes in action
Security 360

Another use of hashes?



Tomorrow!

Security 360