

# Cryptology

codes and ciphers and keys and more

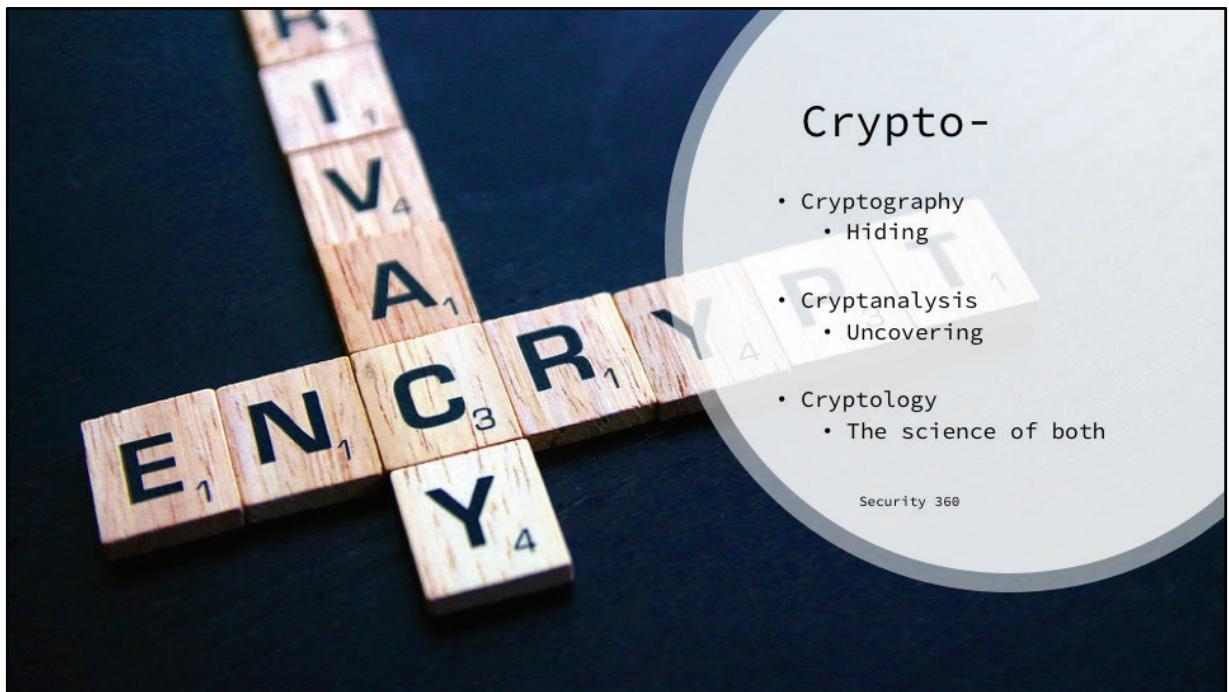
**Can you decode this?**

9 20,8,15,21,7,8,20 19,15

---

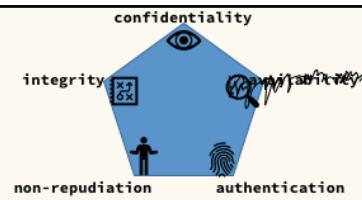
Security 360

I thought so



"Math and Pen" by [theprimaryjosh](#) is licensed under [CC BY-NC-SA 2.0](#)

# 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*

Security 360

## 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
- 

Security 360

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



["Free: Material Design Psd Alphabet"](#) by Crunk C is licensed under [CC BY-NC 4.0](#)



## Reading my first cipher

Step 1: Take your **ciphertext**  
“tfdsfu xpse”

Step 2: Reverse **cipher**, with same **key**  
*<rewind each letter by one>*

Step 3: Get the **plaintext!**  
“secret word”

Security 360

["Free: Material Design Psd Alphabet"](#) by Crunk C is licensed under [CC BY-NC 4.0](#)

# Classical Cryptography

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



Security 360

"Julius Caesar, Act 1st, Sc. 3d. Cæsar: Von Cassius has a lean and hungry look, would he were fatter. Brutus: Feed him on Libby, McNeill & Libby's Cooked Corned Beef. Cæsar: Ay, do so, good Brutus, let us have only men about us that are fat. [front]" by Boston Public Library is licensed under CC BY 2.0

# Cryptography Through Time



Security 360

["Hieroglyphs"](#) by [InvaderXan](#) is licensed under [CC BY-NC-SA 2.0](#)

["Enigma Machine"](#) by [Shiny Things](#) is licensed under [CC BY 2.0](#)

# 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/>

Security 360

# 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



---

Security 360

"Locked" by [swister\\_p](#) is licensed under [CC BY-NC-ND 2.0](#)

# Modern Cryptography

The image consists of two side-by-side screenshots. The left screenshot shows a web browser window with a red exclamation mark icon in the top-left corner. Below it, the text "Your connection is not private" is displayed. A message follows: "Attackers might be trying to steal your information from 178.22.68.112 (for example, passwords, messages or credit cards). [Learn more](#)". Below this is the error code "NET::ERR\_CERT\_AUTHORITY\_INVALID". At the bottom of this window are two buttons: "Advanced" and "Back to safety". The right screenshot shows the "Security & Privacy" preferences pane in Mac OS X. The "FileVault" tab is selected. It displays the text: "FileVault secures the data on your disk by encrypting its contents automatically." Below this is a "WARNING:" message: "You will need your login password or a recovery key to access your files. If you lose your password and recovery key, the data will be lost." To the right of this message is a "Turn Off" button. At the bottom of this pane, it says "FileVault is turned on for the disk 'Macintosh HD'." and "A recovery key has been set.".

["Matrix letter1"](#) by [Adct2Luv](#) is licensed under [CC BY-NC-ND 2.0](#)

# 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”

**FLIP!** ↓ **RESET!**

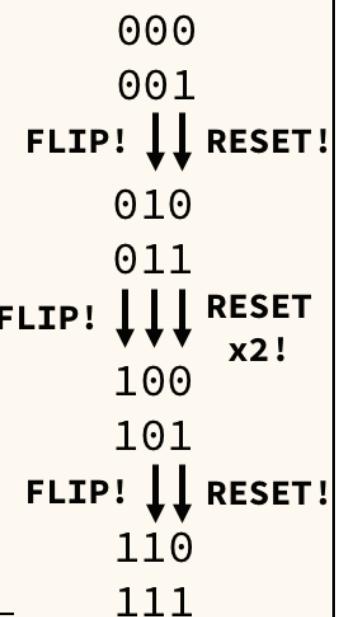
Security 360

**10**

## 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!
- 



Security 360

## Computer Encryption

```
1101010100101010101010101001  
0101010101010101010100110001011  
010010101010101010101001001010101  
010101010101010100110101001010101  
010101010101001001010101010101010  
101010100111101011010010101010101  
010101001001010101010101010101010  
100111010101010101010101010101010  
01001010101010101010101010101110  
101010010101010101010100100101010  
101010101010101010011001010101010
```

Security 360

"@ @" by [dalcrose](#) is licensed under [CC BY-NC-SA 2.0](#)

## A very simple cipher

---

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

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

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

$0 \oplus 0 = 0$

$1 \oplus 0 = 1$

---

Security 360

# Bitwise XOR

---

## Encryption

Plaintext (DEC)	123
Plaintext (BINARY)	1111011
Cipher (XOR)	$\oplus$
Key (BINARY)	0100011
Ciphertext (BINARY)	1011000

## Decryption

Ciphertext (BINARY)	1011000
Cipher (XOR)	$\oplus$
Key (BINARY)	0100011
Plaintext (BINARY)	1111011

---

## XOR in Practice

---

- Any key can be cracked with the plaintext
  - ciphertext  $\oplus$  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 ...

---

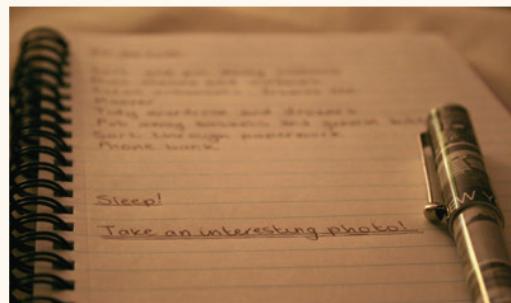
UNBREAKABLE (in theory)

---

Security 360

# 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)



Security 360

["To Do List"](#) by [Mrs Magic](#) is licensed under [CC BY-NC-SA 2.0](#)

## 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>

---

Security 360

<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



Security 360

## Symmetric Encryption

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



Security 360

## Meet Alice & Bob

---



Hi I'm Alice



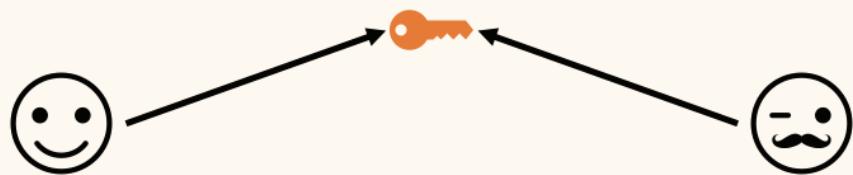
Hi I'm Bob

---

Security 360

## Key Distribution Problem

- How do Alice & Bob share this key?



Security 360

## Key Exchange Problem

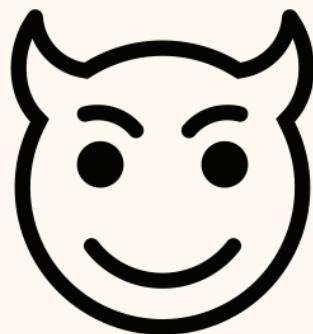
- Can Alice generate the key and send it to Bob?



Security 360

# Introducing Eve

---



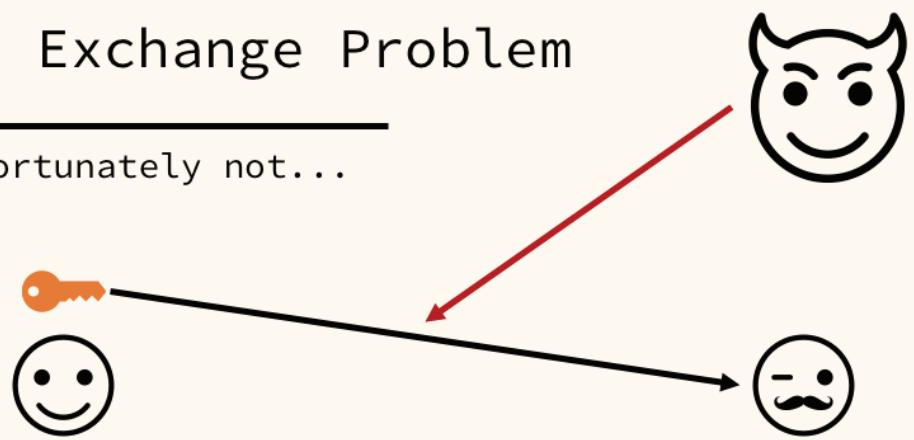
Eve

---

Security 360

## Key Exchange Problem

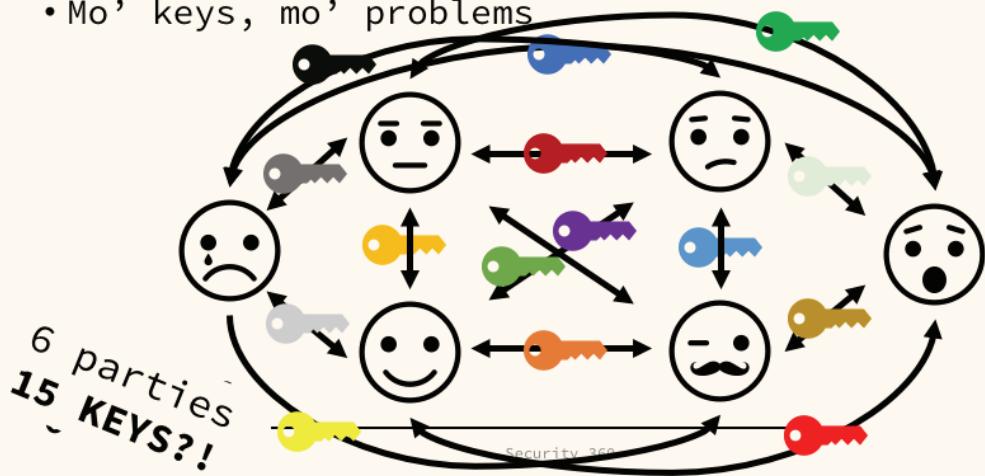
- Unfortunately not...



Security 360

## 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

---

Security 360

## Key Pairs

---

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



---

Security 360

## 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



Security 360

## Key Pairs

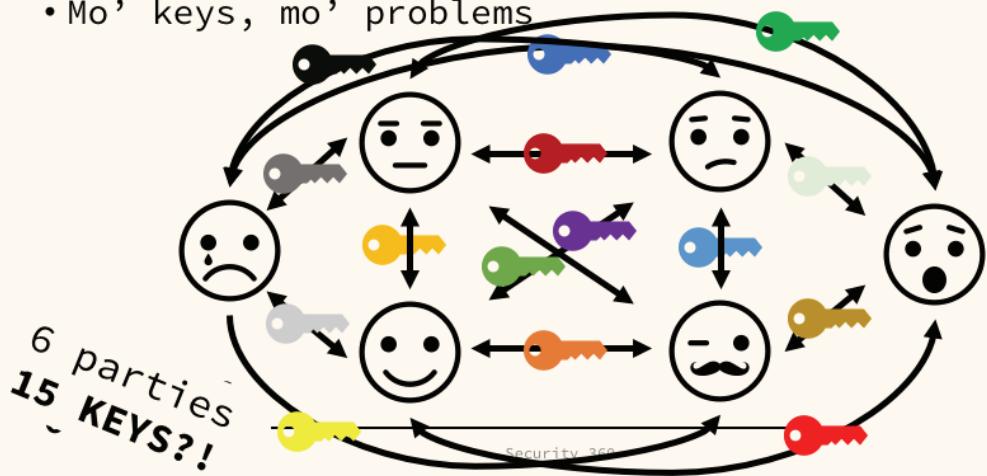
- Others use your public key to encrypt data
- You use your private key to decrypt it!



Security 360

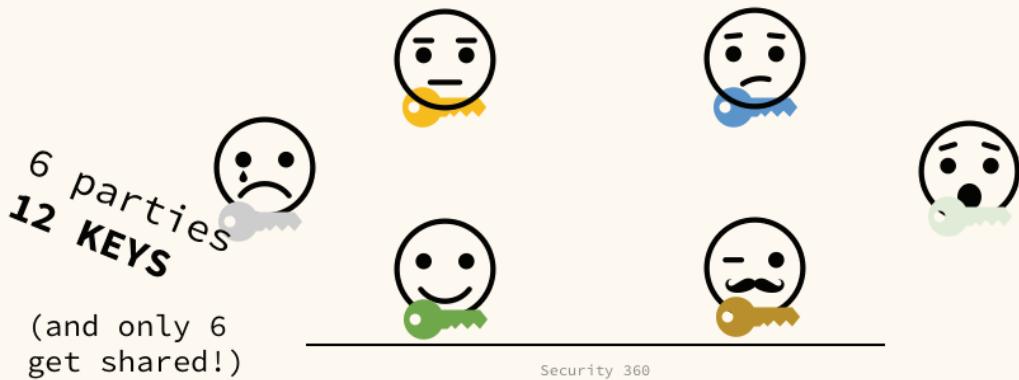
## Remember This?

- Mo' keys, mo' problems



## Asymmetric Saves the Day

- Mo' keys, no' problems



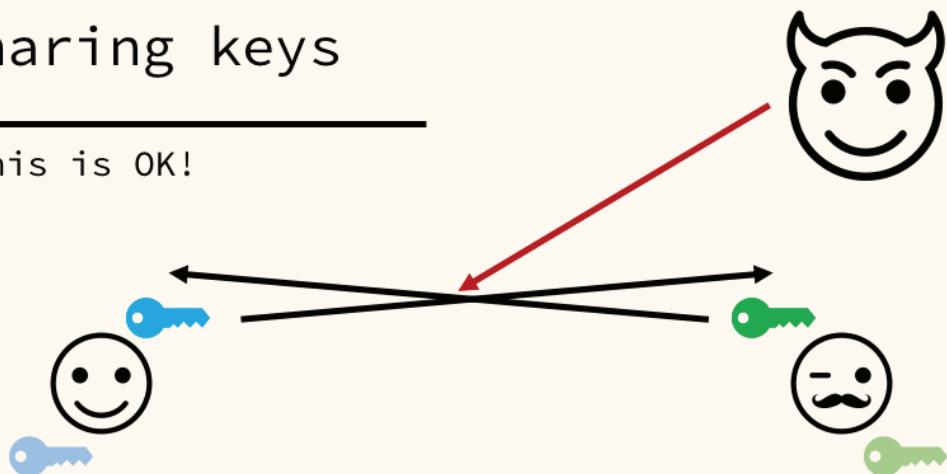
Public  
Keys



Security 360

## Sharing keys

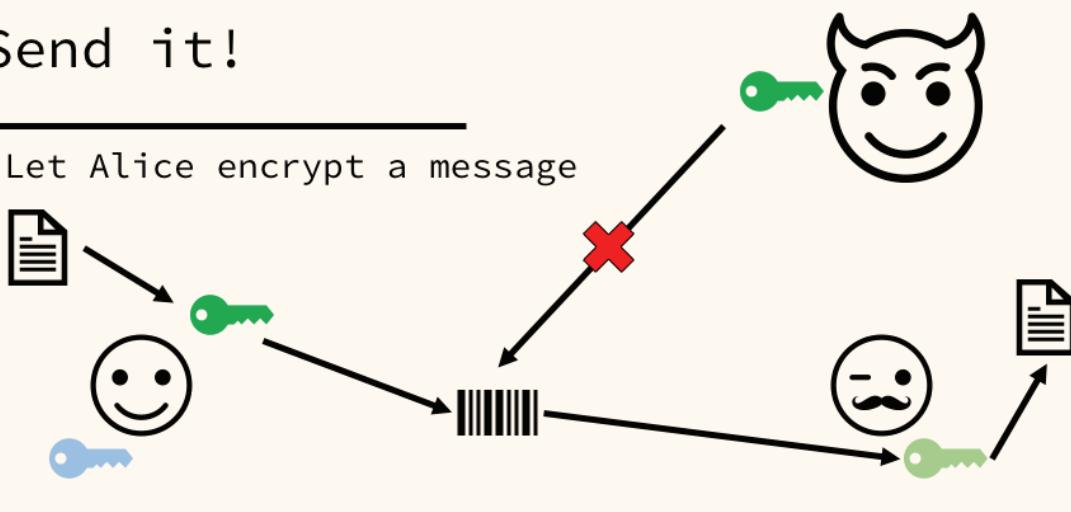
- This is OK!



Security 360

## Send it!

- Let Alice encrypt a message

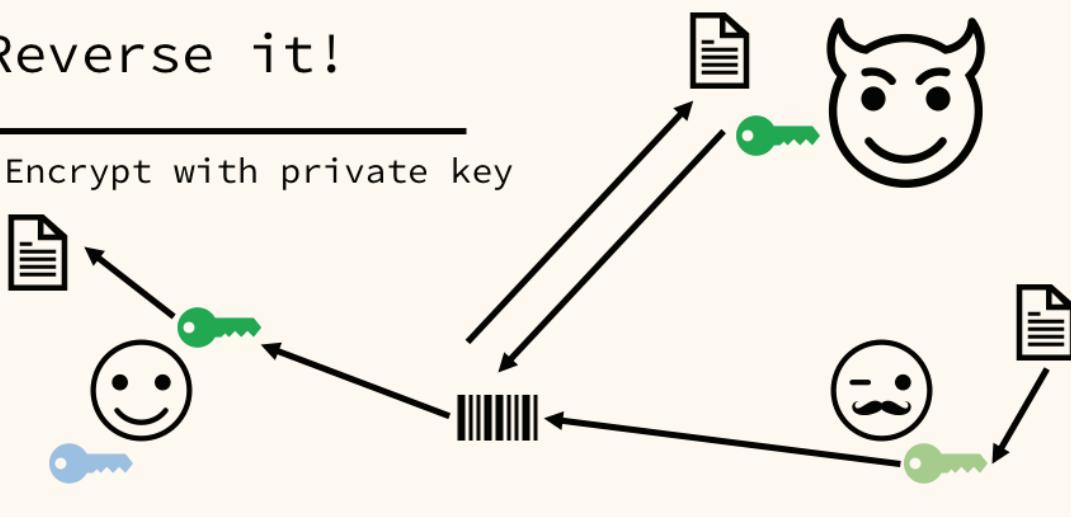


Security 360

Let's try the opposite

## Reverse it!

- Encrypt with private key



Security 360

## But... Why would you do that?

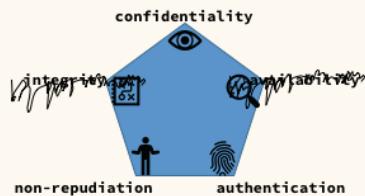
---

- 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
- 

Security 360

## Asymmetric Isn't Perfect

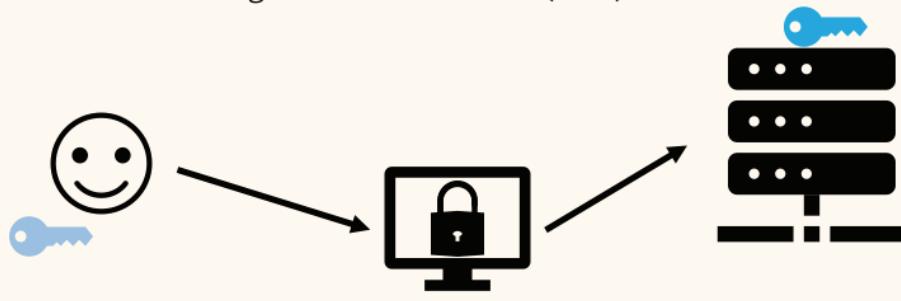
- Slower (less ideal for confidentiality)
- Still breakable (compromised private key)
- How do we know whose public key that is??



Security 360

## A simple example with SSH

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



Security 360

["Turtle in repose"](#) by [thriftyknitter](#) is licensed under [CC BY-NC-SA 2.0](#)

## 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**
- 

Security 360

# Certificates

---

- A way to prove the owner of the public key
- Works on the basis of a **Trusted Third Party**
  - Certificate Authority



Hi I'm Alice



---

Security 360

# Public Key Infrastructure

- Bob trusts that Certificate Authority too!



Hi I'm Alice



Yup that's Alice



Security 360

## In the wild...

- A more familiar scenario



Hi I'm <https://ebay.com>



Yup that's <https://ebay.com>



Security 360

## 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



["P1220166"](#) by [Will.House](#) is licensed under [CC BY-NC-SA 2.0](#)

## Setting Computational 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



Security 360

["Online"](#) by [m104](#) is licensed under [CC BY-NC-SA 2.0](#)

## 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

Volkswagen goes electric.  
The new e-up! is here.  
Inspired by Blue.



Security 360

"Volkswagen goes electric" by Aleksander Erichsen, Thea Emanuelsen is licensed under [CC BY-ND 4.0](#)

# 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!!

Security 360



"One Way" by [Chris Campbell](#) is licensed under [CC BY-NC 2.0](#)

## 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**



---

Security 360

["Air-balloon collision"](#) by [nojhan](#) is licensed under [CC BY-SA 2.0](#)

## 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

"The version I will use." by [trekkyandy](#) is licensed under [CC BY-SA 2.0](#)

# Hashing Passwords

**Sign up**

New user?  
Use the form below to create your account.

First Name: Alice      Last Name: Miller

Nationality: American      E-mail: alice.miller@yahoo.com

Date of Birth: 21 December 1995      Gender: Female

Mobile Number: +1 407-217-6873      OTP: 7241

Password:   
Confirm Password:   
\*Enter the OTP received on your phone.

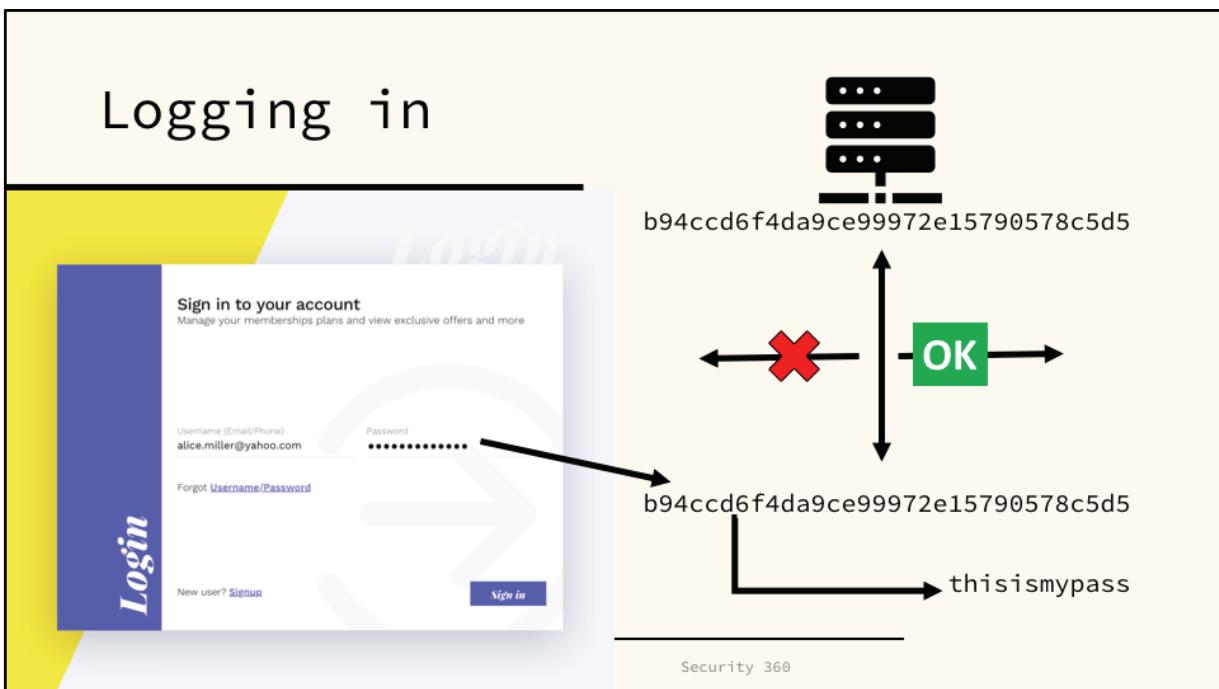
Have an account? [Login](#)

[Complete Signup](#)

b94cccd6f4da9ce99972e15790578c5d5

*"Login and Signup form - Free Download"* by Mauritius D'Silva is licensed under [CC BY-NC-ND 4.0](#)

# Logging in



*"Login and Signup form - Free Download"* by Mauritius D'Silva is licensed under [CC BY-NC-ND 4.0](https://creativecommons.org/licenses/by-nd/4.0/)

# 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:

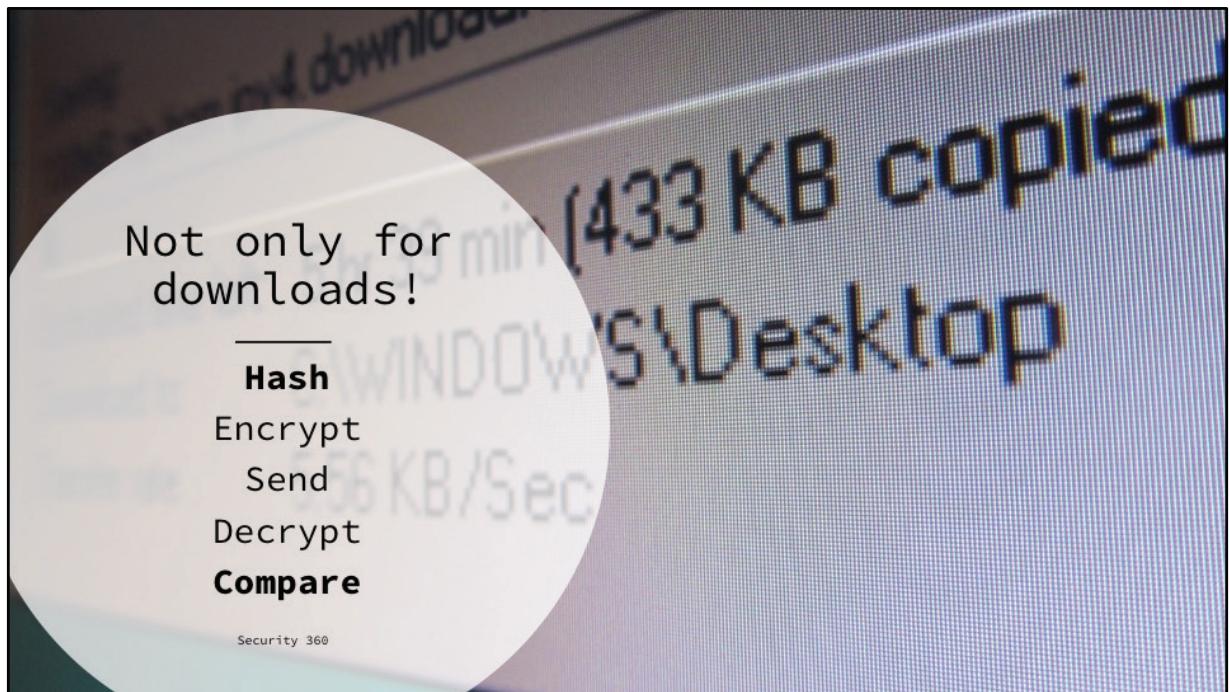
[hello.exe](#). MD5 Sum: cdc47d670159eef60916ca03a9d4a007  
[erase.exe](#). MD5 Sum: cdc47d670159eef60916ca03a9d4a007

## Linux version (i386):

[hello](#). MD5 Sum: da5c61e1edc0f18337e46418e48c1290  
[erase](#). MD5 Sum: da5c61e1edc0f18337e46418e48c1290

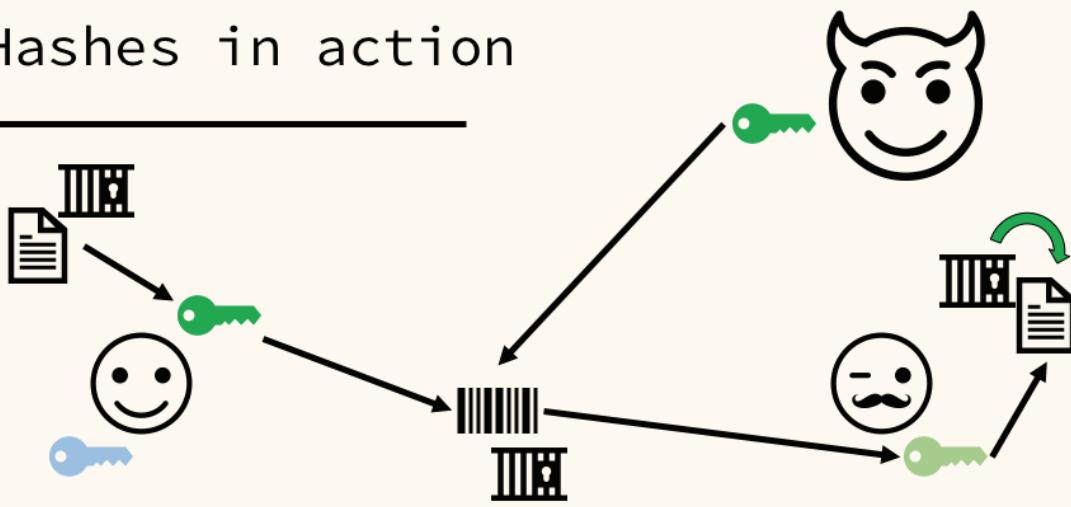
Security 360

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



*"Downloading file"* by [Christiaan Colen](#) is licensed under [CC BY-SA 2.0](#)

## Hashes in action



Security 360

## Another use of hashes?

---



Tomorrow!

---

Security 360

["Bitcoin Illustrations Freebies"](#) by Loredana Papp-Dinea, Mihai Baldean, Milo Themes is licensed under [CC BY-NC-ND 4.0](#)