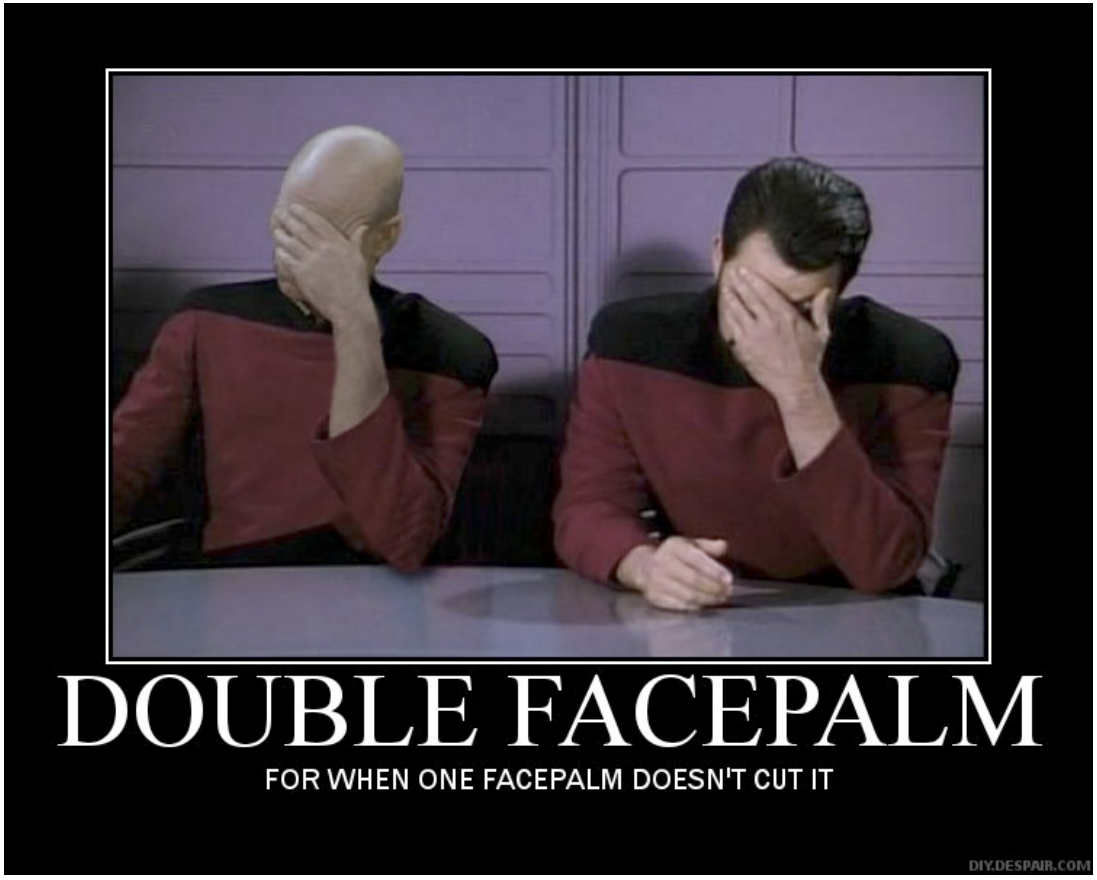
CSCI 3330 Cybersecurity

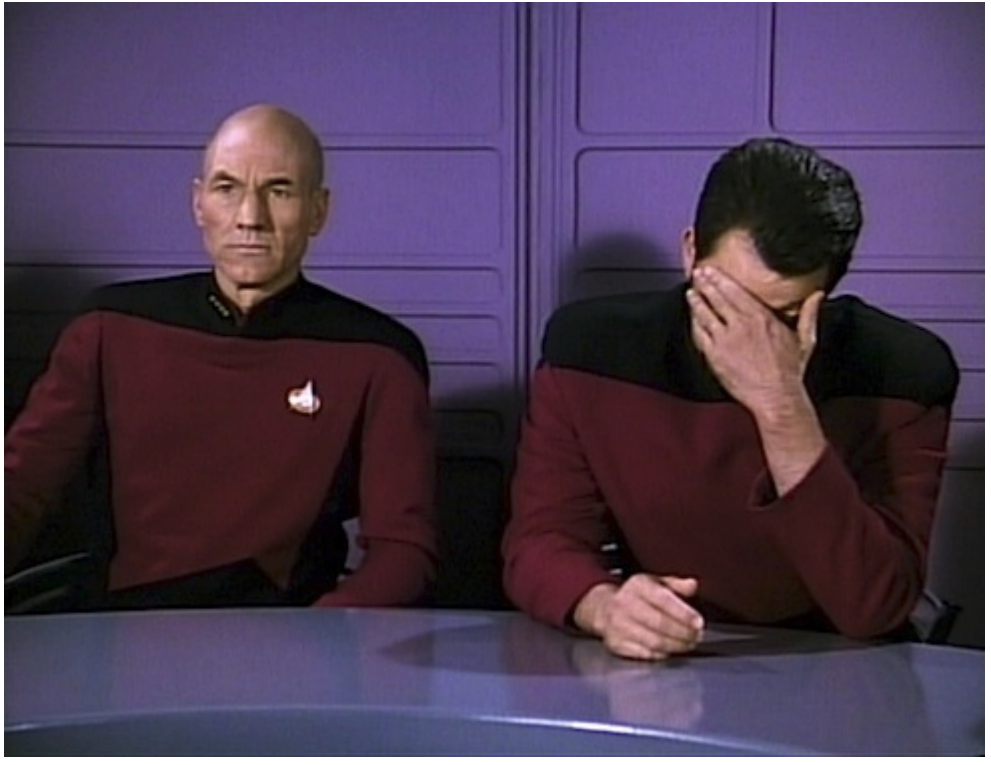


- 6: (1,5) (2,4) (3,3) (3,3) (4,2) (5,1)
- 7: (1,6) (2,5) (3,4) (4,3) (5,2) (6,1)

- If we have (2,4) and (4,2), why not (3,3) and (3,3)?
- There are two paths to 6 with 2.
  - Your first roll can be either 2 or a 4. Then you must roll a 4 or a 2, respectively.
- There is one path to 6 with 3.
  - First, you must roll a 3. Then you must roll a 3.

- 6: (1,5) (2,4) (3,3) (3,3) (4,2) (5,1)
- 7: (1,6) (2,5) (3,4) (4,3) (5,2) (6,1)

- 6: (1,5) (2,4) (3,3) ~~(3,3)~~ (4,2) (5,1)
- 7: (1,6) (2,5) (3,4) (4,3) (5,2) (6,1)



Word of the day

LIKE NAILING JELLY TO A TREE adj. Used to describe a task thought to be impossible, esp. one in which the difficulty arises from poor specification or inherent slipperiness in the problem domain. “Trying to display the ‘prettiest’ arrangement of nodes and arcs that diagrams a given graph is like nailing jelly to a tree, because nobody's sure what ‘prettiest’ means algorithmically.”

Problem

- What if two users have the same password?

Salt

- Store ***NON***-secret string with password
- Concatenate when hashing
- hash($salt . $password)

Salt

- Protects against accidental collisions
  - Birthday paradox!
  - In a group of just 23 people, there’s a greater than 50% chance that at least two of them share the same birthday
- Protects against rainbow table lookups

When does birthday paradox apply?

- Hash collisions?
- Brute-forcing passwords?
- Buying a lottery ticket?
- Two people buying the same lottery ticket?
- Two people buying the winning ticket?

Password hashes

- Let’s look at our real-world example:
- https://brainhammer.com/dt/

Rainbow tables

- Fixed spans (e.g., alphanumeric up to 6 chars)

Why Salting is Important

Prevents Rainbow Table Attacks:

Without salting, attackers could use precomputed "rainbow tables" containing common password hashes to quickly identify users' passwords. A unique salt for each password makes these precomputed tables useless because every hash will be different.

Mitigates Dictionary Attacks:

Dictionary attacks rely on guessing passwords from lists of common words. Since salting adds random characters, a dictionary word will not produce the same hash as its salted counterpart, rendering these lists ineffective.

Protects Against Data Breaches:

If a database is breached, attackers will have access to the salt and the hashed passwords. However, without knowing the specific salt used for each password, they cannot easily determine the original passwords or identify if multiple users share the same password.

Increases Brute-Force Difficulty:

Salting increases the complexity of password cracking, forcing attackers to perform a separate, unique calculation for each salted hash, significantly increasing the time and resources required to guess a password.

Why Salting is Important

Prevents Rainbow Table Attacks:

Without salting, attackers could use precomputed "rainbow tables" containing common password hashes to quickly identify users' passwords. A unique salt for each password makes these precomputed tables useless because every hash will be different.

Mitigates Dictionary Attacks:

Dictionary attacks rely on guessing passwords from lists of common words. Since salting adds random characters, a dictionary word will not produce the same hash as its salted counterpart, rendering these lists ineffective.

Protects Against Data Breaches:

If a database is breached, attackers will have access to the salt and the hashed passwords. However, without knowing the specific salt used for each password, they cannot easily determine the original passwords or identify if multiple users share the same password.

Increases Brute-Force Difficulty:

Salting increases the complexity of password cracking, forcing attackers to perform a separate, unique calculation for each salted hash, significantly increasing the time and resources required to guess a password.

Cracking passwords

- John the Ripper
- hashcat
- Rainbow tables

Breaking cryptographic hashes

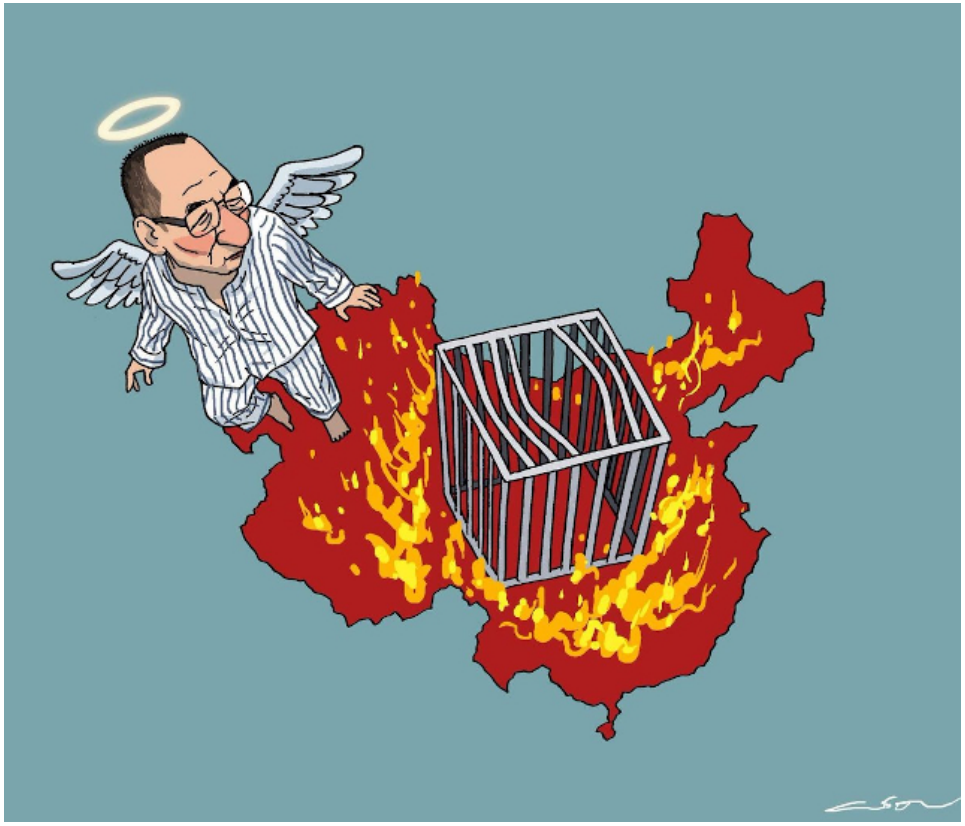
- Wang Xiaoyun
- MD5 fails collision resistance
- SHA-1 if you have ~$100k

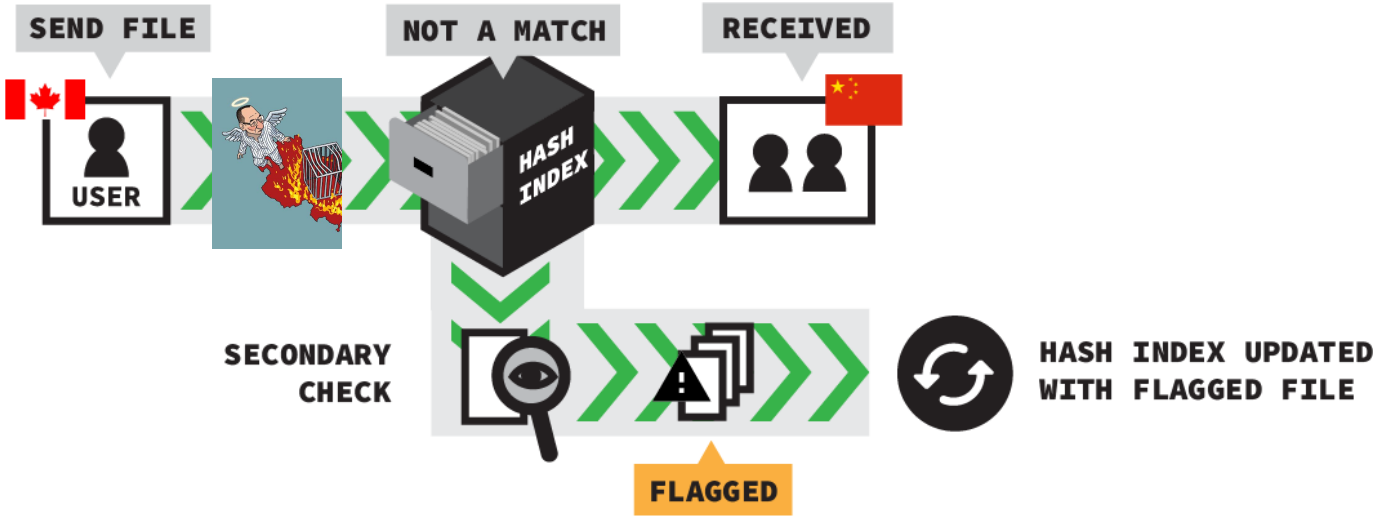
Hashes are used for signing

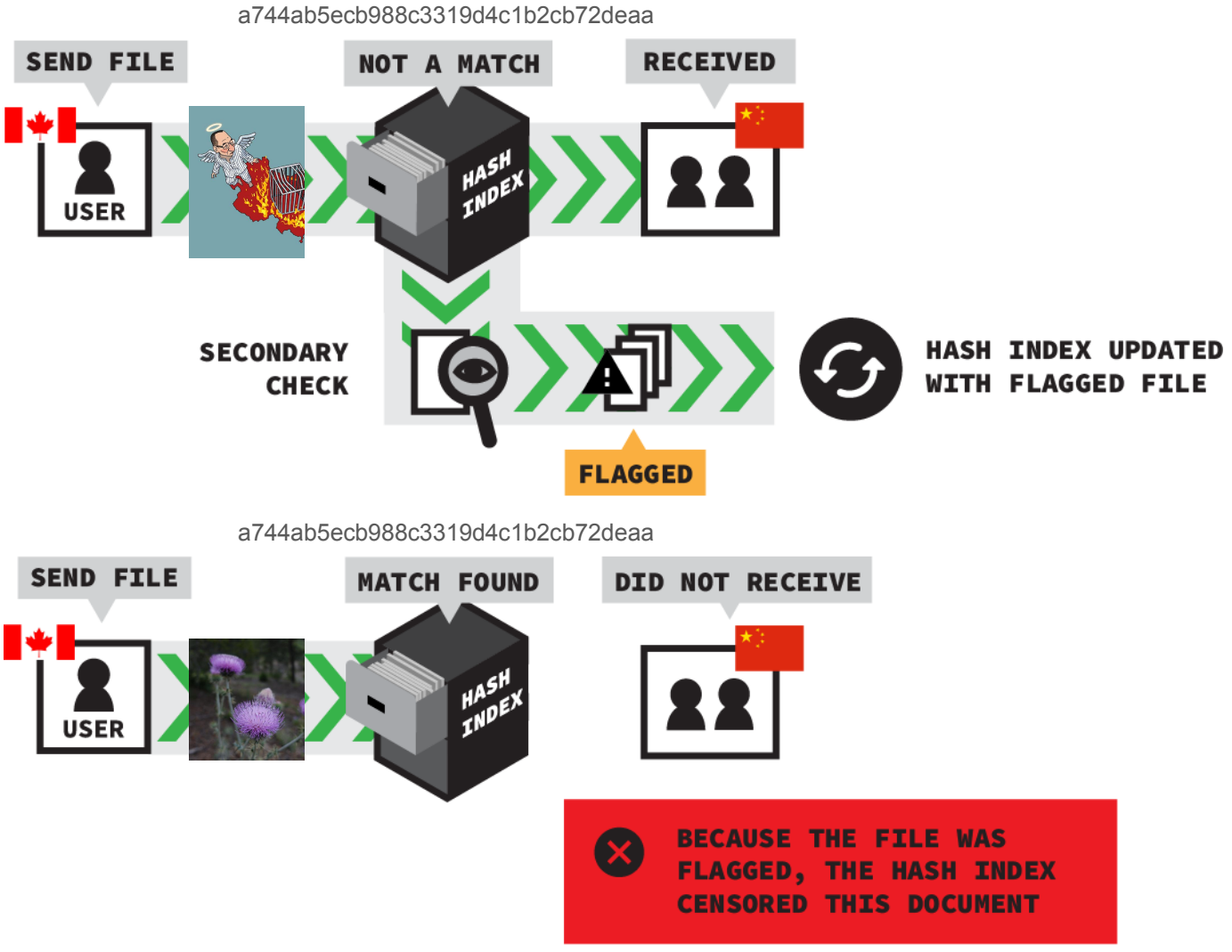
- Digitally signing is computationally expensive
- Just sign the hash
- Fool someone into signing something else

Most downloaded apps in 2023?

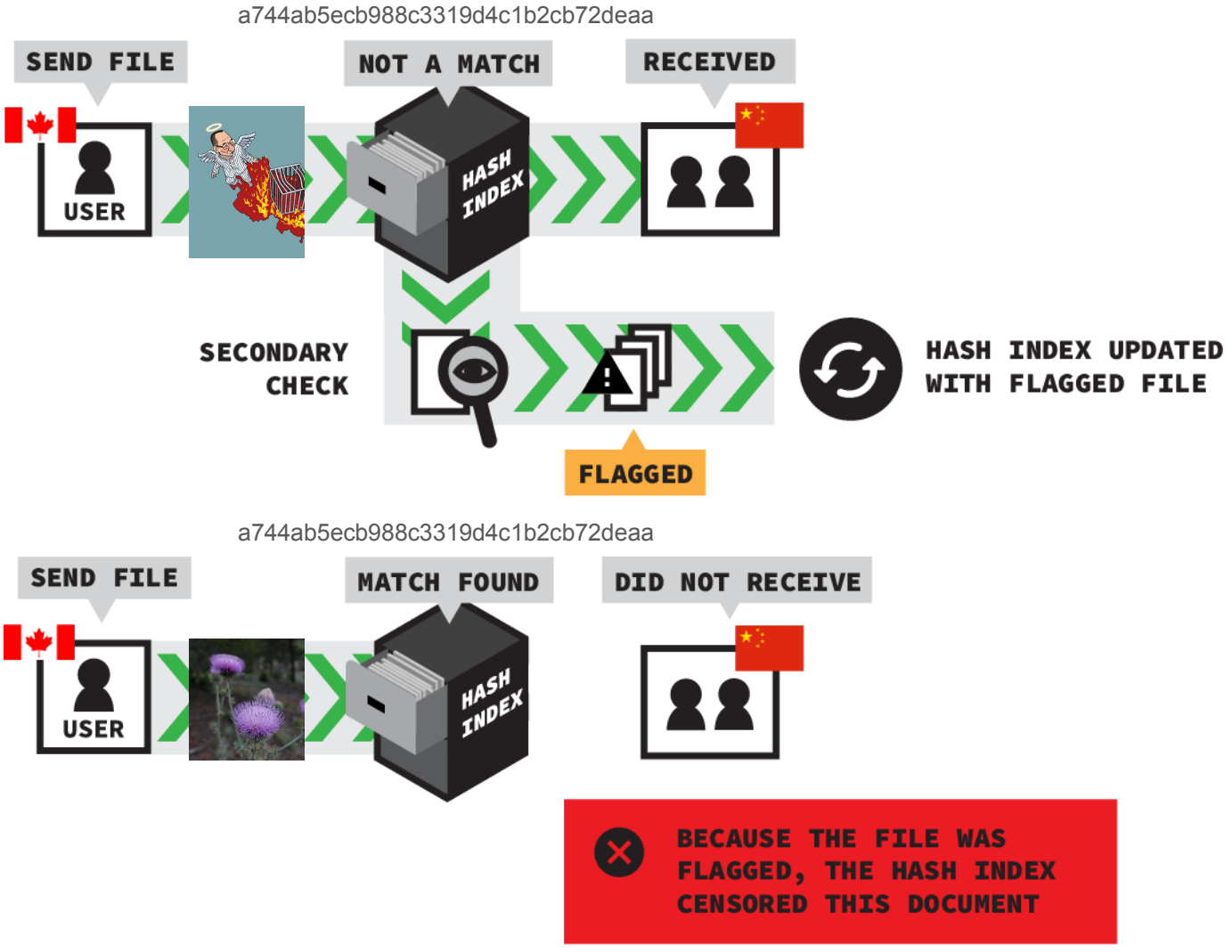

WeChat

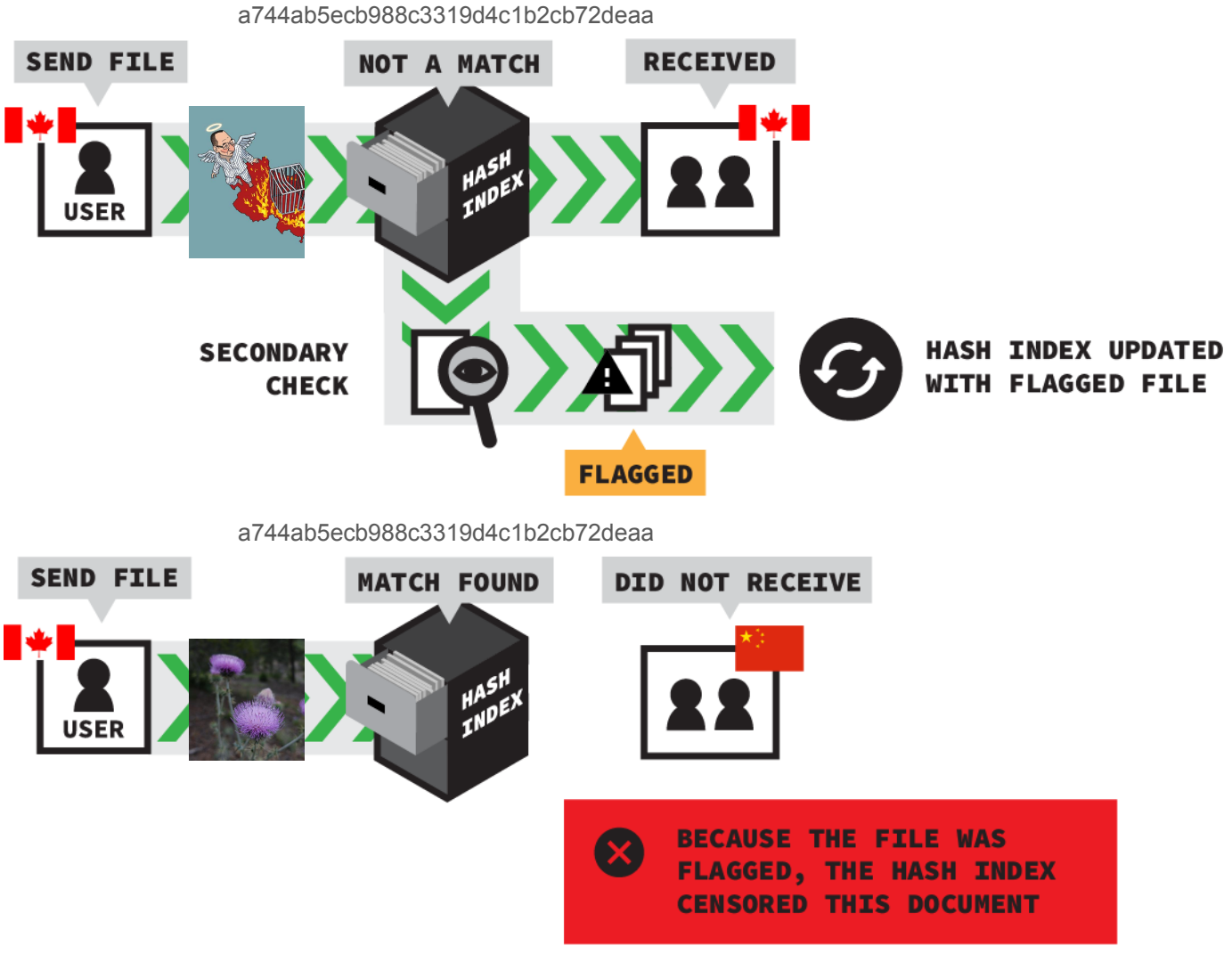



SEND FILE
NOT A MATCH
RECEIVED
USER
HASH INDEX
SECONDARY CHECK
FLAGGED
HASH INDEX UPDATED WITH FLAGGED FILE


a744ab5ecb988c3319d4c1b2cb72deaa
SEND FILE
NOT A MATCH
RECEIVED
USER
HASH INDEX
SECONDARY CHECK
HASH INDEX UPDATED WITH FLAGGED FILE
FLAGGED
a744ab5ecb988c3319d4c1b2cb72deaa
SEND FILE
MATCH FOUND
DID NOT RECEIVE
USER
HASH INDEX
BECAUSE THE FILE WAS FLAGGED, THE HASH INDEX CENSORED THIS DOCUMENT

- WeChat censors Chinese users on WeChat
- WeChat does ***NOT*** censor non-Chinese users
- Is WeChat safe to use for non-Chinese users?


a744ab5ecb988c3319d4c1b2cb72deaa
SEND FILE
NOT A MATCH
RECEIVED
USER
HASH INDEX
SECONDARY CHECK
HASH INDEX UPDATED WITH FLAGGED FILE
FLAGGED
a744ab5ecb988c3319d4c1b2cb72deaa
SEND FILE
MATCH FOUND
DID NOT RECEIVE
USER
HASH INDEX
BECAUSE THE FILE WAS FLAGGED, THE HASH INDEX CENSORED THIS DOCUMENT


a744ab5ecb988c3319d4c1b2cb72deaa
SEND FILE
NOT A MATCH
RECEIVED
USER
HASH INDEX
SECONDARY CHECK
HASH INDEX UPDATED WITH FLAGGED FILE
FLAGGED
a744ab5ecb988c3319d4c1b2cb72deaa
SEND FILE
MATCH FOUND
DID NOT RECEIVE
USER
HASH INDEX
BECAUSE THE FILE WAS FLAGGED, THE HASH INDEX CENSORED THIS DOCUMENT

- Tencent has since ceased this surveillance affecting > 100 million users.

THE WALL STREET JOURNAL.

SUBSCRIBE

SIG

TECH

China's WeChat Monitors For Users to Refine Censorship at

While they aren't c
through the app ar

The Washington Post

Sign in

Tech Help Desk Artificial Intelligence Internet Culture Spa

Chinese censorship invades the U.S. via WeChat

Users in North America say the app has blocked them
ntent displeasing to Chinese
e support Trump's ban effort, which will
f a court hearing Jan. 14.

FINANCIAL TIMES

Tencent Holdings Ltd + Add to myFT

Tencent monitors foreign accounts to aid domestic censorship

Research finds WeChat analyses messages from users registered outside China to train its algorithms

Informed policymaking

- Congressional staffers
- U.S. Executive Order

FOR OFFICIAL USE ONLY

UNITED STATES DEPARTMENT OF COMMERCE
Office of Intelligence and Security
Deputy Assistant Secretary for Intelligence and Security
Washington, D.C. 20230

September 17, 2020

MEMORANDUM FOR THE SECRETARY

THROUGH: Rob Blair
Director
Office of Policy and Strategic Planning

FROM: John K. Costello
Deputy Assistant Secretary for Intelligence and Security
Office of Intelligence and Security

SUBJECT: Proposed Prohibited Transactions Related to WeChat Pursuant to Executive Order 13943

- Computer scientists have immense power
- Cybersecurity experts even more
- Hackers even more
- Use it for good, not evil
- For more: Philip Rogaway's *The Moral Character of Cryptographic Work*

setuid root

- You can only set your effective UID to either your real UID or your effective UID
- So how do su and sudo work?
- Must be some way to become root again!

setuid bit (and gid…)

- When you execute a file, if the setuid bit is set, your effective ID becomes that of the file's owner
- When the file is owned by root, we call this a *setuid root* binary

```
ls -l /usr/bin
...
-rwxr-sr-x 1 root crontab       39664 Mar 30  2024 crontab
...
-rwsr-xr-x 1 root root          55680 Jun  5 08:17 su
-rwsr-xr-x 1 root root         277936 Jun 25 08:42 sudo
...
```

- su, sudo: become superuser indefinitely if you pass some test
- crontab drops privileges to edit

Command line injection

- In-band signals
- Where else have we seen these?

- `printf '%d %d == %d\n' 1 2 3`
- `... steps ...`
- `main() with argv:`

  `{“printf”, “%d %d == %d\n”, “1”, “2”, “3”}`
- `... steps ...`
- `1 2 == 3`

Shell constructs argv

- Backslashes and quoting
  - → Files with *spaces*
- `execv(argv)`

Conventions

- `ls -l -a == ls -la`
- `wget -O page.html mysite.com`
  `== wget -Opage.html mysite.com`
- `ls -l -l -l == ls -lll == ls -l`
- `wget -Ox.html -Oy.html mysite.com`
  `== wget -Oy.html mysite.com`

Assignment 1

- Deadline: September 18, 2025, 10:00pm
- Permissions, entropy, randomness
- https://tildesites.bowdoin.edu/~j.knockel/csci3330f25/assignment1.pdf