CMPT 980 – Information Privacy

Schedule: M 3:30 – 4:20 pm, Th 2:30 – 4:20 pm

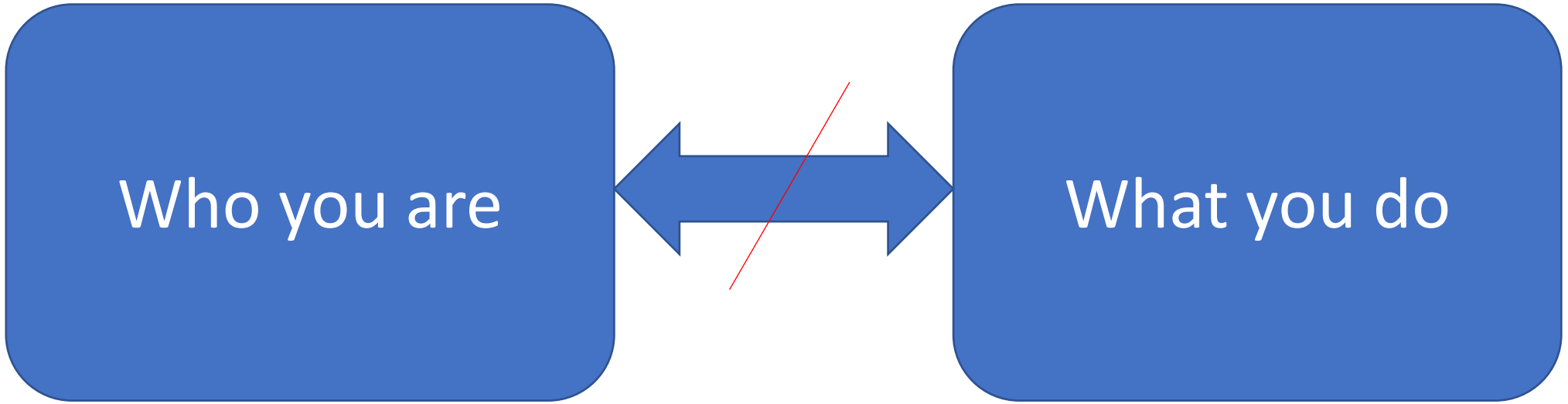
Feel free to write to me at any time: (my full name, no spaces) xxxxxxx@sfu.ca, and please preface your e-mail with “CMPT 980”.

CMPT 980 – Information Privacy

Grading:

- 10% Participation
  - These will be self-assessment quizzes on Canvas
- 30% Paper Reading
  - 5 deadlines for paper reviews (every 2 weeks)
- 30% Course Project
  - Short Paper OR Software+Demo
- 30% Final Exam
  - Multiple-choice exam on Canvas

What is privacy?



- Privacy is the disassociation between your *identity* and your *behavior*
- It is the right to self-expression

The “nothing to hide” argument (Solove ‘07)

<u>Why do we need privacy if we have nothing to hide?</u>

“If you’ve got nothing to hide, you’ve got nothing to fear”

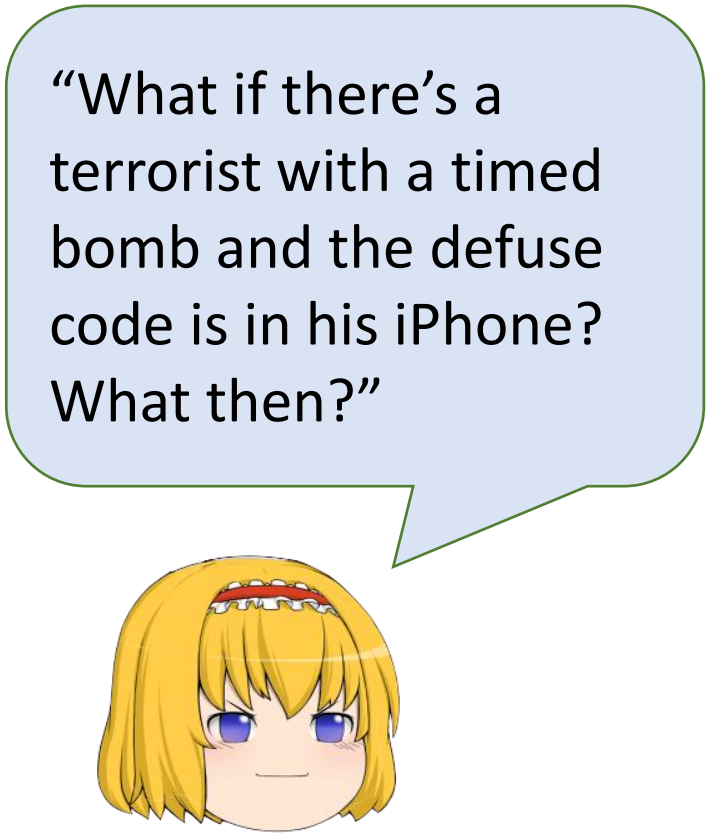
- The easy (and bad) retort: Why do we have curtains?
- More difficult questions:
  - Should the law protect people in concealing discreditable information?
  - Is it an exchange of a small amount of privacy for a large amount of security?
  - Is there a tension between the individual and society?
  - What is the social value of privacy?

What is privacy?

Belief: Privacy comes at the cost of security

Problems:

- We need to examine whether or not scenarios in which privacy and security oppose each other are arbitrarily constructed
- More often than not, the same technological solutions enhance both privacy and security
- Solutions to compromise privacy for security usually involve trusting a party that is immune to attacks – that is inherently insecure

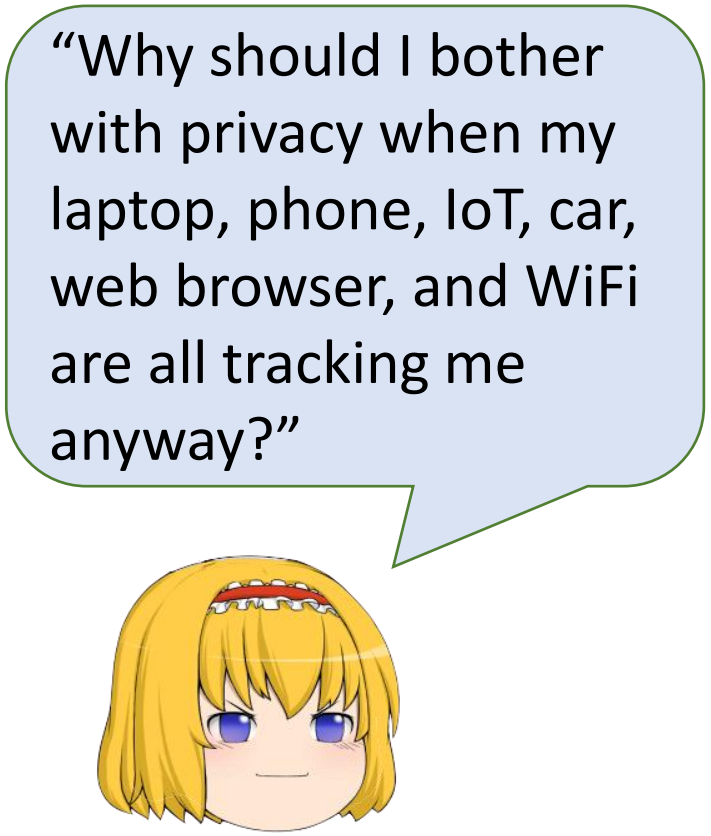


What is privacy?

Belief: There is no point in protecting privacy because we already leak information about ourselves everywhere

Problems:

- A few small steps can significantly reduce one's information leakage
- Improvements in technology and public policy are being made constantly
- Don't let perfect be the enemy of good



What is privacy?

Belief: Privacy is bad because it damages businesses

Problems:

- The collection of user information is *not* a compromise of privacy when it is done with informed consent and the user has control over its retention
- Requiring businesses to divulge data collection and usage practices should not be controversial
- Explicit promises on proper data usage help build up trust between customer and business

Covid-19 Contact Tracing

Privacy & Goals

Goals:

- Warning people if someone they had contact with recently was infected so they can quarantine and/or be tested
- Saying **who** was infected is not a goal

Privacy concerns:

- Sharing someone's health information compromises their privacy
- Sharing someone's location also compromises their privacy
  - It would be nice if we told people where they were infected, but that is not possible considering this concern

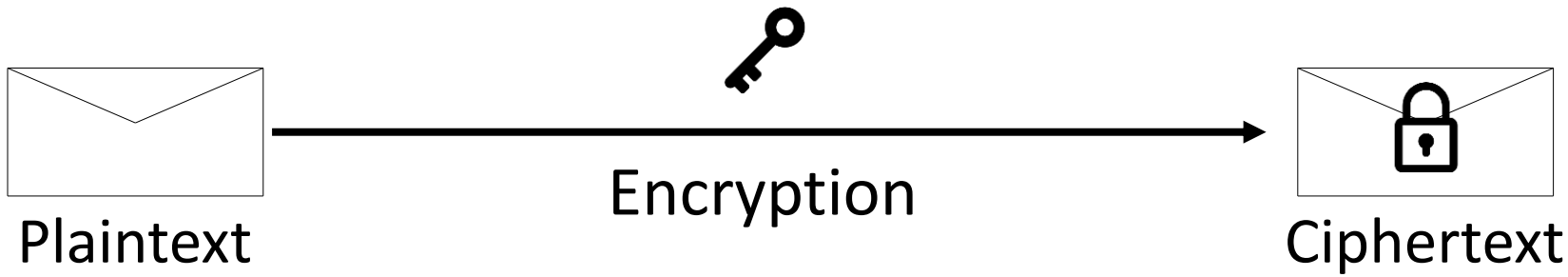
Threat Model

- When you come into contact with someone, they will need to know that in case you test positive later
- When you are infected, you will have to announce it in a public way; this message can be read by anyone
- It is tempting to use a trusted third-party – but this is not a privacy-enhancing technology solution

Cryptographic primitives

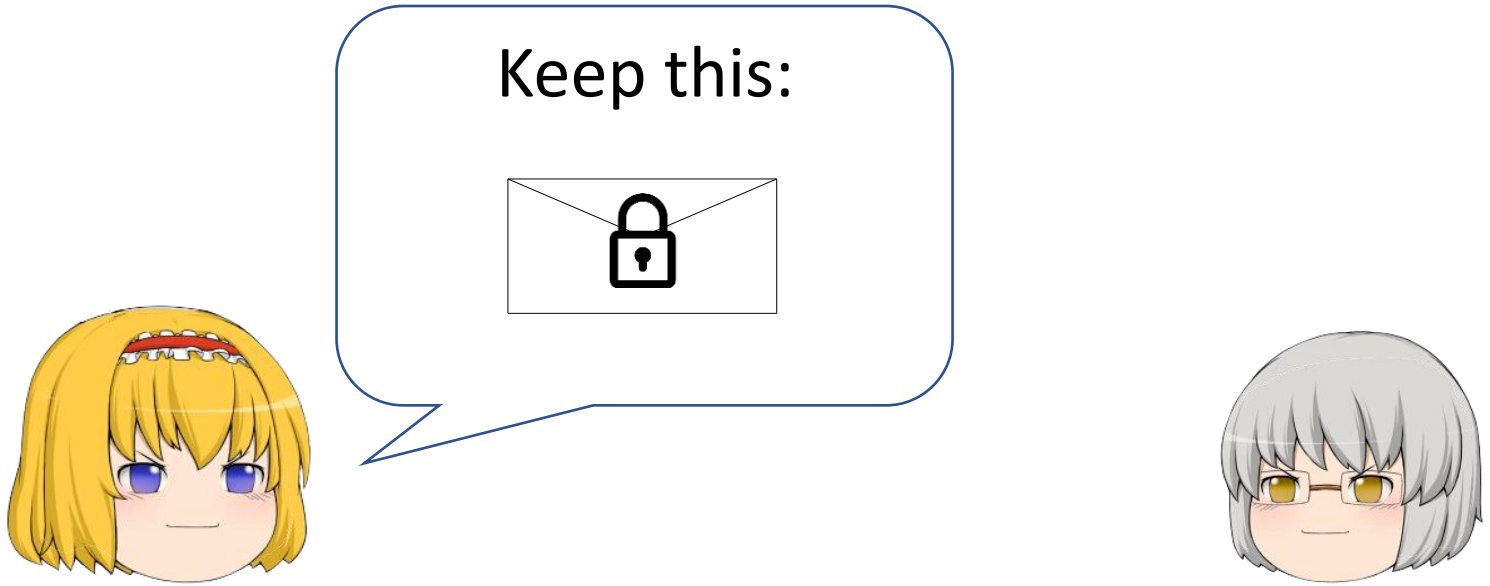
A cryptosystem consists of:

- Key(s)
- Encryption mechanism
- Decryption mechanism



Idea

- I can announce my presence to nearby devices as *ciphertext*
  - The ciphertext is passed through Bluetooth
  - My actual location is never announced
- Only when I am infected will I release the key



Protocol

- Generate a new *temporary exposure key* (K) every day
- When in range of anyone, send them a *rolling proximity identifier* through Bluetooth encrypted as:

$$Enc_K(\text{Known string})$$

- Since we want them to be able to decrypt this later using the same key, it should be a symmetric encryption (e.g. AES)
- Retain up to 14 temporary exposure keys on your device

https://covid19.apple.com/contacttracing

Protocol

- If testing positive, release all 14 temporary exposure keys to a public database
- All users regularly check the public database and attempt to decrypt the RPIs they have stored using those keys
- If a decryption attempt is successful, you will also know how many days ago you came into contact with that person
  - Success = known string observed

Problems?

- Storage – a few hundred million keys is only gigabytes
- Computation – most computation is done on individual devices, and thousands of AES decryptions per second is no problem
- Information leakage – my RPI doesn't reveal anything about me, not does my key
- But what if an eavesdropper listens to RPIs across several locations to try to track people?
  - The fact that the RPI doesn't change in a day is an undesirable property

$$Enc_K(\text{Known string})$$

Increasing RPI granularity

- Based on the temporary exposure key K, derive *rolling keys* RK every 10 minutes
  - This can be done with key derivation functions (a hash)

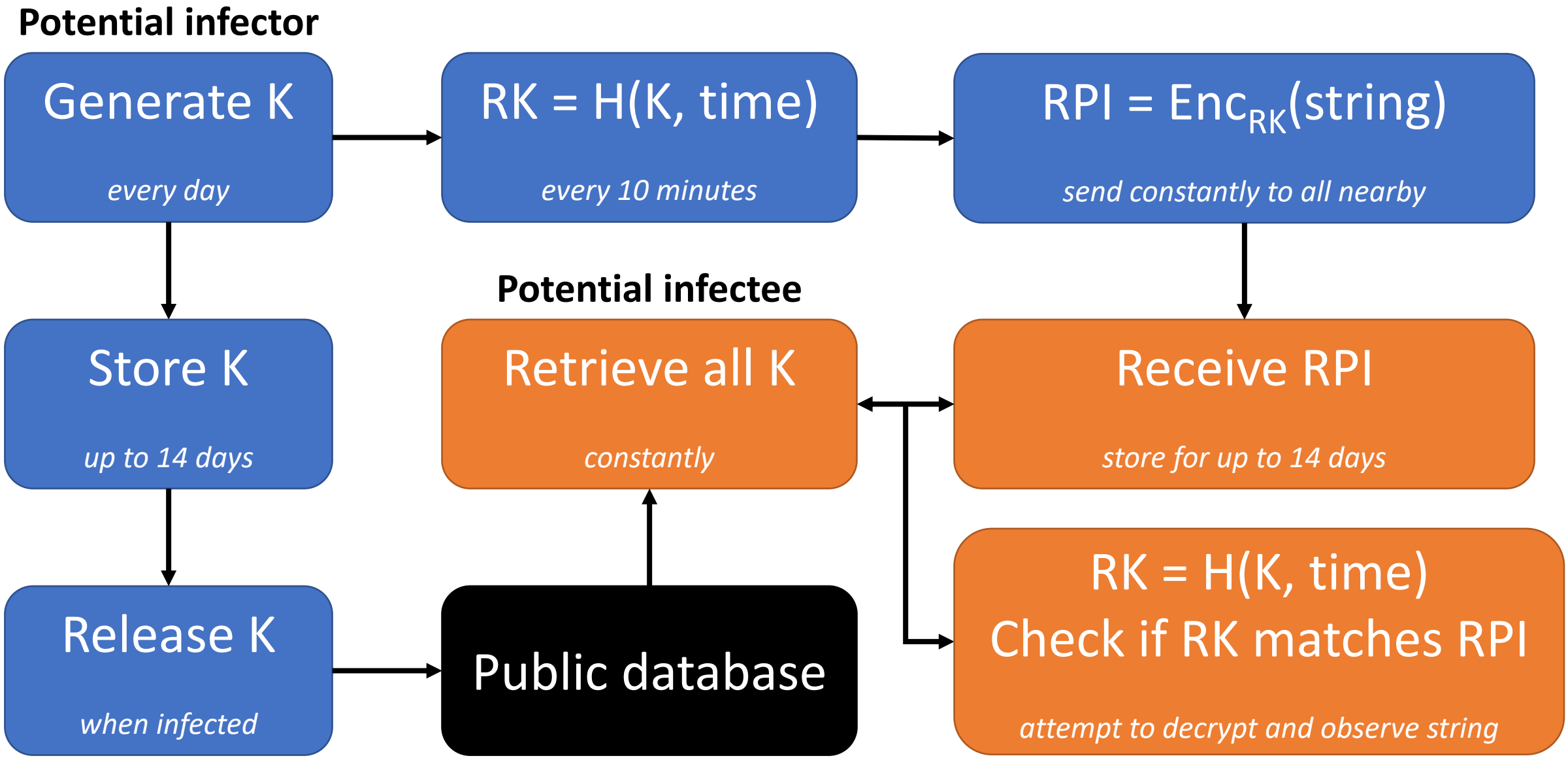
$$RK = H(K, time)$$

- Instead of encrypting with K, encrypt with RK

$$Enc_{RK}(Known\ string)$$

- Still release K when testing positive; recipients compute RK
- Could we have used RK in the first place?
  - Yes; but this trick reduces transfer/storage by 144 times

Overview


Potential infector
Generate K
every day
RK = H(K, time)
every 10 minutes
RPI = Enc_RK(string)
send constantly to all nearby
Store K
up to 14 days
Release K
when infected
Potential infectee
Retrieve all K
constantly
Public database
Receive RPI
store for up to 14 days
RK = H(K, time)
Check if RK matches RPI
attempt to decrypt and observe string