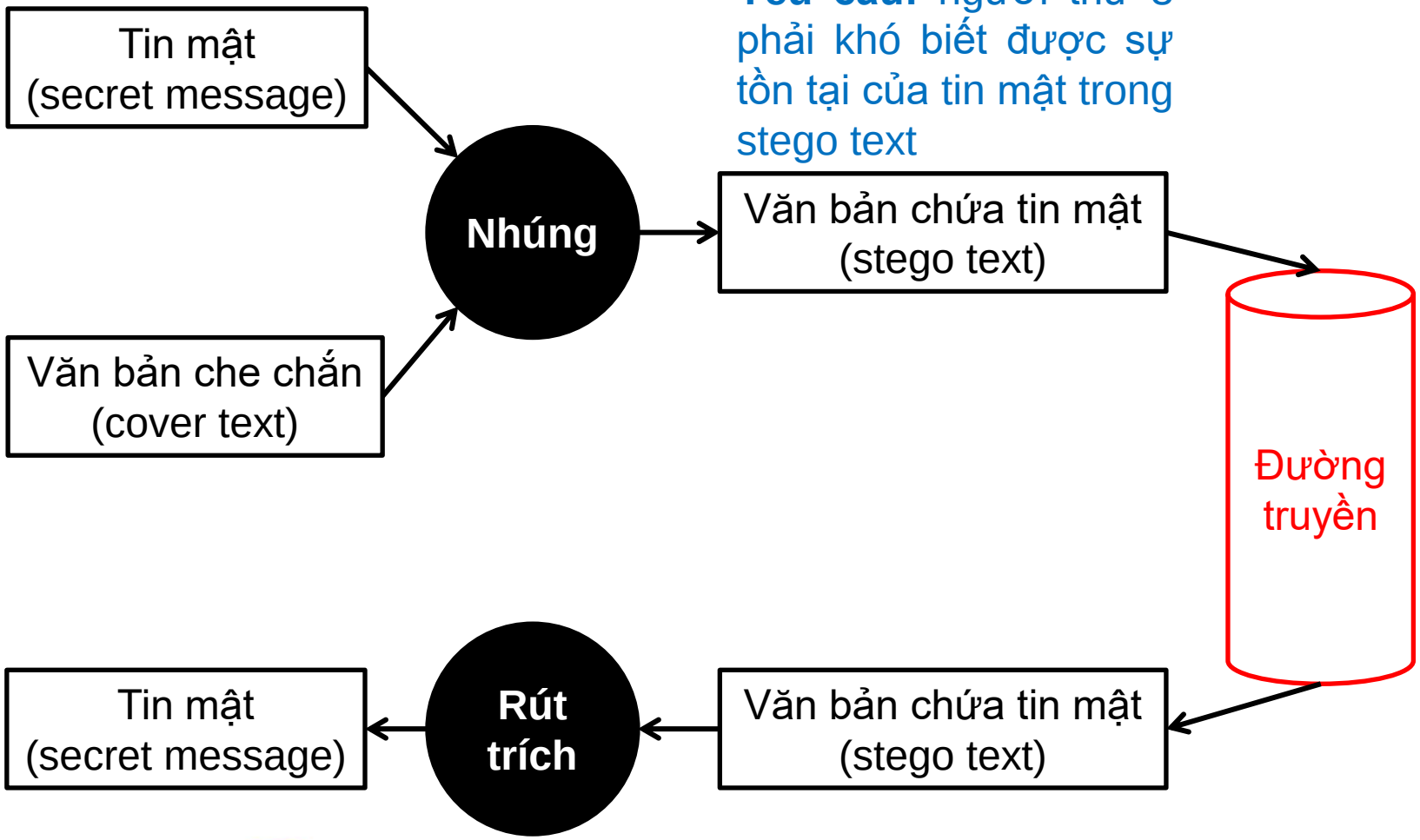
Ẩn tin mật trên văn bản (Text steganography)

KHOA CÔNG NGHỆ THÔNG TIN
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

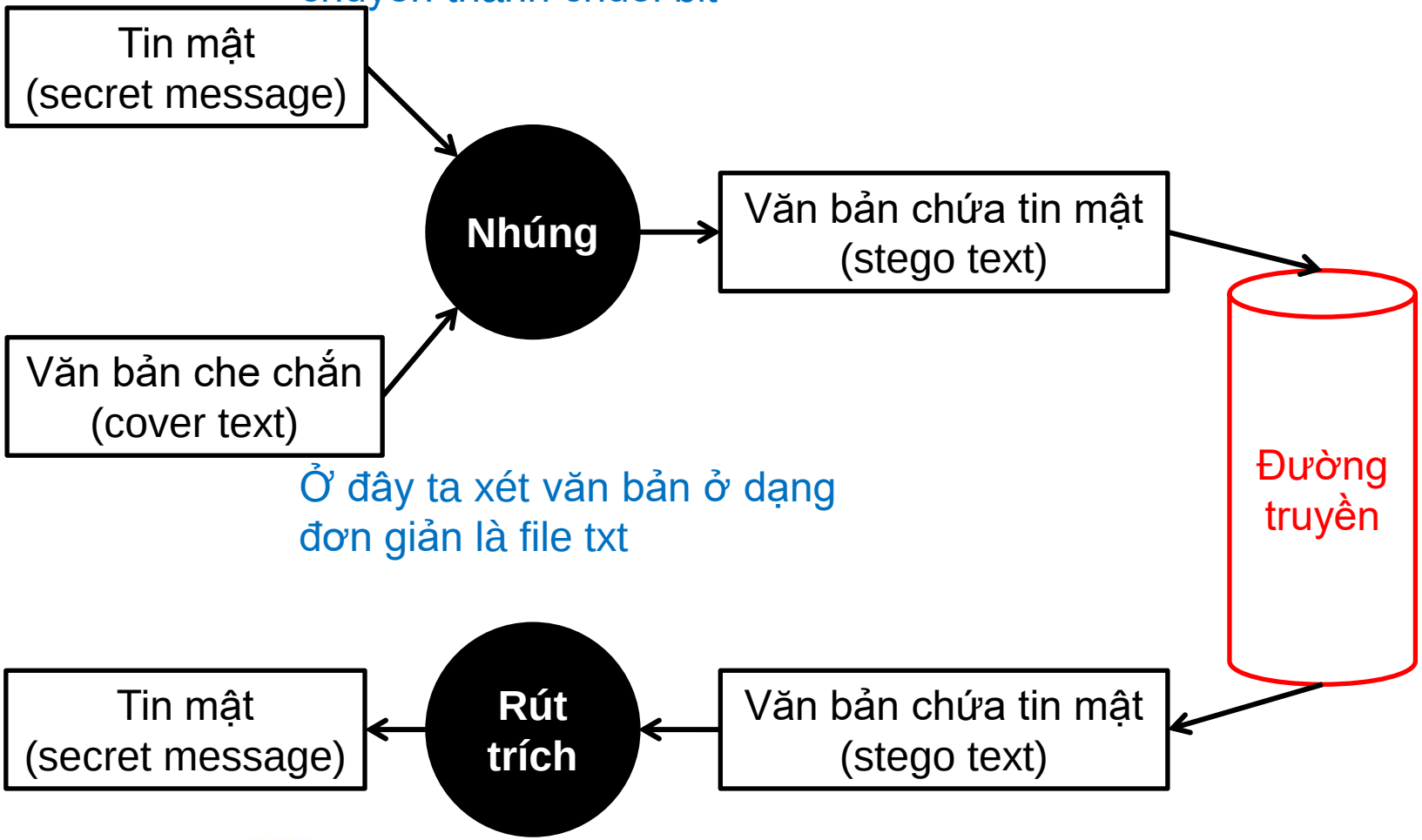
Bài toán ẩn tin mật trên văn bản

Yêu cầu: người thứ 3 phải khó biết được sự tồn tại của tin mật trong stego text



Bài toán ẩn tin mật trên văn bản

Tin mật thường sẽ được chuyển thành chuỗi bit



Ở đây ta xét văn bản ở dạng đơn giản là file txt

Nhóm phương pháp dùng khoảng trắng của văn bản

- Ví dụ: bit 0 = 1 khoảng trắng, bit 1 = 2 khoảng trắng
- Dùng khoảng trắng sẽ không làm thay đổi ngữ nghĩa của văn bản che chắn ☺
- Nên dùng khoảng trắng ở đâu để văn bản nhìn vẫn bình thường?

Phương pháp dùng khoảng trắng cuối mỗi dòng

Ý tưởng

- Ở cuối mỗi dòng, nếu muốn nhúng bit 0 thì chèn 1 khoảng trắng, bit 1 thì chèn 2 khoảng trắng, không nhúng thì không chèn khoảng trắng
- Thường yêu cầu chiều dài của dòng sau khi nhúng phải ≤ một con số nào đó (vd, 70) → có những dòng sẽ không nhúng được

```
Hello Allice,  bit 1
 bit 0
Have you heard about steganography? I find it very interesting and  bit 1
want to share with you. Below is the introduction from Wiki. bit 0
  bit 1
Steganography is the practice of concealing a file, message, image, or  Ko nhúng
video within another file, message, image, or video. The word bit 0
steganography combines the Greek words steganos, meaning "covered,  Ko nhúng
concealed, or protected," and graphein meaning "writing".  Ko nhúng
```

Phương pháp dùng khoảng trắng cuối mỗi dòng

Nhận xét

- Tính vô hình?
  - ☺
- Tính bền vững?
  - Một số trình xử lý văn bản có thể tự động xóa các khoảng trắng cuối mỗi dòng ☹
- Sức chứa?
  - Chứa được khoảng số dòng bit ☹
  - Một số dòng sau khi nhúng vẫn còn có chỗ để chèn thêm khoảng trắng → có thể tận dụng điều này để tăng sức chứa lên không?
    - Một cách là nhúng nhiều hơn một bit ở mỗi dòng; vd, nhúng 2 bit ở mỗi dòng: 00 = 1 khoảng trắng, 01 = 2 khoảng trắng, 10 = 3 khoảng trắng, 11 = 4 khoảng trắng
    - Có đảm bảo tăng sức chứa so với nhúng 1 bit ở mỗi dòng?

Phương pháp dùng khoảng trắng giữa các từ để làm cho văn bản được căn lề 2 bên

Ý tưởng

- Giữa các từ có rất nhiều khoảng trắng, tại sao không dùng để nhúng bit (vd, bit 0 = 1 khoảng trắng, bit 1 = 1 khoảng trắng **chèn thêm 1 khoảng trắng**)?
  - Tính vô hình ☹
- Để văn bản được căn lề 2 bên, việc thêm một số khoảng trắng vào giữa các từ là điều bình thường → tận dụng điều này để nhúng bit
- Nhưng không thể làm đơn giản là: bit 0 = 1 khoảng trắng, bit 1 = 1 khoảng trắng **chèn thêm 1 khoảng trắng**
  - Vì chịu ràng buộc về số khoảng trắng phải chèn cho mỗi dòng
  - Với ví dụ ở dưới, dòng 1 phải chèn 0 khoảng trắng → giả sử muốn nhúng bit 1 thì không nhúng được, khi rút trích làm sao biết dòng này không có bit nhúng?

```
Steganography is the practice of concealing a file, message, image, or<- Phải chèn 0 khoảng trắng
video within another file, message, image, or video. The word·········<- Phải chèn 9 khoảng trắng
steganography combines the Greek words steganos, meaning "covered,····<- Phải chèn 4 khoảng trắng
concealed, or protected," and graphein meaning "writing".
```

Phương pháp dùng khoảng trắng giữa các từ để làm cho văn bản được căn lề 2 bên

Tất nhiên, ta chỉ làm với dòng căn lề (xem ví dụ ở hình bên dưới)

Ý tưởng

Một đề xuất là, với mỗi dòng:

- Bit 0 = 10 = 1 khoảng trắng **chèn thêm 1 khoảng trắng** + 1 khoảng trắng
- Bit 1 = 01 = 1 khoảng trắng + 1 khoảng trắng **chèn thêm 1 khoảng trắng**
- Không nhúng = các trường hợp còn lại

```
Hello Allice,

Have you heard about steganography? I find it very interesting and    <- Dòng căn lề
want to share with you. Below is the introduction from Wiki.

Steganography is the practice of concealing a file, message, image, or<- Dòng căn lề
video within another file, message, image, or video. The word         <- Dòng căn lề
steganography combines the Greek words steganos, meaning "covered,    <- Dòng căn lề
concealed, or protected," and graphein meaning "writing".
```

Phương pháp dùng khoảng trắng giữa các từ để làm cho văn bản được căn lề 2 bên

Ví dụ 1: nhúng chuỗi bit 101 vào cover text ở dưới

- Dòng 1: không nhúng được. Khi rút trích, gặp 1 khoảng trắng + 1 khoảng trắng → biết ngay là dòng này không được nhúng
- Dòng 2: có 9 khoảng trắng và phải chèn thêm 9 khoảng trắng → không nhúng được, mỗi khoảng trắng sẽ được chèn thêm một khoảng trắng. Khi rút trích, gặp 2 khoảng trắng + 2 khoảng trắng → biết ngay là dòng này không được nhúng
- Dòng 3: có 7 khoảng trắng và phải chèn 4 khoảng trắng → nhúng được 3 bit. Khi rút trích, gặp 1 khoảng trắng + 2 khoảng trắng → bit 1, gặp 2 khoảng trắng + 1 khoảng trắng → bit 0; còn nếu gặp trường hợp khác 2 trường hợp này → dừng

cover text

```
Steganography is the practice of concealing a file, message, image, or<- Phải chèn 0 khoảng trắng
video within another file, message, image, or video. The word         <- Phải chèn 9 khoảng trắng
steganography combines the Greek words steganos, meaning "covered,    <- Phải chèn 4 khoảng trắng
concealed, or protected," and graphein meaning "writing".
```

stego text

```
Steganography is the practice of concealing a file, message, image, or
video  within  another  file,  message,  image,  or  video.  The  word
steganography combines  the  Greek words steganos,  meaning  "covered,
concealed, or protected," and graphein meaning "writing".
```

bit 1 bit 0 bit 1

Phương pháp dùng khoảng trắng giữa các từ để làm cho văn bản được căn lề 2 bên

Ví dụ 2: vẫn với cover text ở ví dụ 1, nhưng chuỗi bit nhúng là 10

- Ở dòng 3 có thể nhúng 3 bit nhưng chuỗi bit nhúng chỉ có 2 bit
- Sau khi nhúng 2 bit, còn 3 khoảng trắng và phải chèn 2 khoảng trắng → phải làm sao để có thể rút trích được?
  - Một cách là: 2 khoảng trắng + 2 khoảng trắng + 1 khoảng trắng
- Nhưng ngoài ví dụ này, vẫn có những trường hợp khác; vd, ở một dòng, sau khi nhúng hết bit, còn 2 khoảng trắng và phải chèn 1 khoảng trắng → phải làm sao để có thể rút trích được???
  - Một cách giúp mọi thứ đơn giản hơn là làm sao để luôn có bit để nhúng: sau khi nhúng hết chuỗi bit, ta sẽ nhúng: một bit 1, còn lại là bit 0
    - Khi rút trích, làm bình thường. Chuỗi bit rút trích được sẽ gồm chuỗi bit nhúng + đuôi 100...; ta có thể dễ dàng cắt đuôi 100... này

cover text

```
Steganography is the practice of concealing a file, message, image, or<- Phải chèn 0 khoảng trắng
video within another file, message, image, or video. The word·····<- Phải chèn 9 khoảng trắng
steganography combines the Greek words steganos, meaning "covered, ···<- Phải chèn 4 khoảng trắng
concealed, or protected," and graphein meaning "writing".
```

Phương pháp dùng khoảng trắng giữa các từ để làm cho văn bản được căn lề 2 bên

Thuật toán nhúng

Input:

- msg_bits: chuỗi bit ứng với tin mật
- cover_text: văn bản dùng để che chắn tin mật
- text_width: độ dài của dòng sau khi được căn lề

Output:

- stego_text: là cover_text sau khi đã được nhúng msg_bits

Phương pháp dùng khoảng trắng giữa các từ để làm cho văn bản được căn lề 2 bên

Thuật toán nhúng

- b = 0 # Chỉ số duyệt từng bit của msg_bits
- l = 0 # Chỉ số duyệt từng dòng của cover_text
- Lặp trong khi mà l < số dòng của cover_text
  - Nếu dòng l là dòng căn lề # Có thể nhúng được bit
    - n = số khoảng trắng của dòng l
    - m = số khoảng trắng cần phải chèn để dòng l được căn lề = text_width – len(dòng l)
    - Nếu 0 < m < n # Nhúng được min(m, n-m) bit
      - Duyệt min(m, n-m) cặp khoảng trắng đầu của dòng l
        - Nếu b < len(msg_bits) thì nhúng msg_bits[b] vào cặp khoảng trắng theo qui ước: bit 0 = 1 khoảng trắng **chèn thêm 1 khoảng trắng** + 1 khoảng trắng, bit 1 = 1 khoảng trắng + 1 khoảng trắng **chèn thêm 1 khoảng trắng**; b += 1
        - Ngược lại: lần đầu (trong cả quá trình nhúng) nhúng bit 1, những lần sau nhúng bit 0
      - Với các khoảng trắng còn lại của dòng l: nếu min(m, n-m) = n-m thì chèn thêm một khoảng trắng vào mỗi khoảng trắng
    - Còn nếu m ≥ n: không nhúng được bit, nhưng mỗi khoảng trắng cần được chèn thêm ít nhất một khoảng trắng để dòng l được căn lề
  - l += 1
- Nếu vẫn chưa nhúng được bit 1 (trong đuôi 100...): NHÚNG THẤT BẠI!

Phương pháp dùng khoảng trắng giữa các từ để làm cho văn bản được căn lề 2 bên

Thuật toán rút trích

Input: stego_text; output: extracted_msg_bits

- extracted_msg_bits = rỗng
- l = 0 # Chỉ số duyệt từng dòng của stego_text
- Lặp trong khi mà l < số dòng của stego_text
  - Nếu dòng l là dòng căn lề
    - Duyệt các khoảng trắng của dòng l
      - Nếu gặp 2 khoảng trắng + 1 khoảng trắng: thêm bit 0 vào extracted_msg_bits
      - Còn nếu gặp 1 khoảng trắng + 2 khoảng trắng: thêm bit 1 vào extracted_msg_bits
      - Ngược lại: không duyệt tiếp nữa
  - l += 1
- Cắt đoạn đuôi 100.. ra khỏi extracted_msg_bits

Phương pháp dùng khoảng trắng giữa các từ để làm cho văn bản được căn lề 2 bên

Thử paste nội dung ở file stego_text vào gmail để gửi cho Alice ...

Tại sao trong gmail nội dung của stego_text không được căn lề hai bên?

Do font chữ mặc định cho phép các ký tự khác nhau có độ rộng khác nhau

Một font chữ trong gmail mà các ký tự đều có cùng độ rộng là Fixed Width

Phương pháp dùng khoảng trắng giữa các từ để làm cho văn bản được căn lề 2 bên

Nhận xét

- Tính vô hình?
  - ☺
- Tính bền vững?
  - Một số trình xử lý văn bản có thể tự động chuẩn hóa khoảng trắng giữa các từ ☹
- Sức chứa?
  - Chứa được khoảng số dòng × min(m, n-m) bit với m là số khoảng trắng phải chèn để dòng được căn lề và n là số khoảng trắng của dòng

Nhóm phương pháp dùng khoảng trắng của văn bản

Nhận xét chung: nhược điểm của nhóm phương pháp này là không bền vững khi bị định dạng lại (một số trình xử lý văn bản tự động chuẩn hóa khoảng trắng)

Hai nhóm phương pháp kế tiếp sẽ khắc phục nhược điểm này và cũng có thể được dùng cùng lúc với nhóm phương pháp khoảng trắng

Nhóm phương pháp dùng cú pháp của văn bản

- Ý tưởng: có những dạng cú pháp có thể dùng thay thế lẫn nhau mà không làm ảnh hưởng đến đến ngữ nghĩa văn bản → bit 0 = dạng này, bit 1 = dạng kia
- Ví dụ
  - Trước từ “and” dùng dấu phẩy hay không dùng dấu phẩy đều được (vd, “text, image, and audio” hay “text, image and audio” đều được) → bit 0 = không có dấu phẩy trước “and”, bit 1 = có dấu phẩy trước “and”
  - Có những cụm từ viết đầy đủ hay viết tắt đều được (“I am” hay “I’m” đều được) → bit 0 = dạng viết tắt, bit 1 = dạng viết đầy đủ
  - ...
- Cẩn thận tính vô hình; vd, dùng dấu phẩy trước từ “and” không nhất quán có thể làm người đọc cảm thấy bất thường

Nhóm phương pháp dùng ngữ nghĩa của văn bản

- Ý tưởng: có những từ đồng nghĩa và có thể dùng thay thế lẫn nhau → bit 0 = từ này, bit 1 = từ kia
- Ví dụ
  - 0 = “big”, 1 = “large”
  - 0 = “little”, 1 = “small”
  - 0 = “chilly”, 1 = “cool”
  - ...
- Nếu có nhiều từ đồng nghĩa thì ta có thể nhúng hơn một bit; vd, với 4 từ đồng nghĩa ta có thể nhúng 2 bit: 00 = từ 1, 01 = từ 2, 10 = từ 3, 11 = từ 4
- Cẩn thận tính vô hình; vd, trong ngữ cảnh khen một người là “cool” thì thay “cool” bằng “chilly” sẽ làm ảnh hưởng đến ngữ nghĩa

Tài liệu tham khảo

W Bender et al., Techniques for data hiding, IBM systems journal (1996)