Chia sẻ thông tin mật (phần 2)

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

Buổi trước: chia sẻ tin mật theo phương pháp Shamir

Phân chia thông tin mật

- Input
  - Thông tin mật S (một con số)
  - Số phần cần chia n
  - Ngưỡng k
- Quá trình thực hiện
  - Tạo hàm f là một đa thức bậc $k-1$ sao cho $f(0) = S$
  - $S_1 = (1, f(1)), S_2 = (2, f(2)), S_3 = (3, f(3)), \ldots$

Tái tạo thông tin mật

- Input
  - Ngưỡng k
  - n' phần của thông tin mật ($n' \geq k$)
- Quá trình thực hiện
  - Tái tạo hàm f (đa thức bậc $k-1$) từ k phần trong n' phần
  - $S = f(0)$

Buổi này

- Phương pháp Shamir với trường hữu hạn: giải quyết vấn đề biểu diễn số thực không chính xác và tràn số khi tính toán trên máy tính
- Phương pháp Shamir mở rộng cho bài toán chia sẻ ảnh mật

Demo về vấn đề biểu diễn số thực không chính xác và tràn số khi tính toán trên máy tính …

Một giải pháp: dùng trường hữu hạn

- Một trường (field) là một tập hợp mà trên đó:
  - Các phép cộng, trừ (cộng với số ngược dấu), nhân, chia (nhân với số nghịch đảo) được định nghĩa (kết quả của những phép tính này cũng phải nằm trong tập hợp của trường → có tiềm năng để tránh vấn đề tràn số)
  - Và có tính chất như những phép tính trên số thực
- Số nguyên có phải là một trường?
  - Không, vì có những số nguyên khi chia cho nhau thì kết quả không phải là số nguyên
- Số thực là một trường có vô số phần tử
  - Máy tính chỉ có thể biểu diễn được một số lượng hữu hạn phần tử ☹
- Có trường nào có hữu hạn số phần tử và đủ để máy tính có thể biểu diễn không ;-)

Trường nào là trường hữu hạn?

- Xét tập hợp $\mathbb{Z}_{100}$ gồm 100 số nguyên 0, 1, …, 99 với các phép toán được định nghĩa như sau:
  - a cộng b: $a + b$ mod 100
  - a nhân b: $a \times b$ mod 100
  - a trừ b: a cộng $-b$ với $-b \in \mathbb{Z}_{100}$ và $-b$ cộng $b = 0$
    (định nghĩa tương đương: $a - b$ mod 100)
  - a chia b: a nhân b^{-1} với $b^{-1} \in \mathbb{Z}_{100}$ và b^{-1} nhân $b = 1$
- $\mathbb{Z}_{100}$ có phải là một trường?
  - Số nghịch đảo của 3?
    67
  - Số nghịch đảo của 5?
    Không có → $\mathbb{Z}_{100}$ không phải là một trường

Trường nào là trường hữu hạn?

- Người ta đã chứng minh được: $\mathbb{Z}_p$ là một trường khi và chỉ khi p là số nguyên tố
- Dùng trường $\mathbb{Z}_p$ cho phương pháp của Shamir sẽ tránh được vấn đề tính toán không chính xác trên máy tính
  - Chọn p đủ nhỏ để máy tính có thể biểu diễn được p phần tử của trường
  - Chọn p đủ lớn để tránh vấn đề trường có quá ít phần tử, kẻ tấn công có thể thử từng phần tử
  - Tin mật S và số phần n phải là số nguyên $\in [0, p)$
- Thật ra, trong bài báo gốc của Shamir là dùng trường hữu hạn $\mathbb{Z}_p$

Demo cài đặt phương pháp của Shamir với trường hữu hạn …

Buổi này

- Phương pháp Shamir với trường hữu hạn: giải quyết vấn đề biểu diễn số thực không chính xác và tràn số khi tính toán trên máy tính
- Phương pháp Shamir mở rộng cho bài toán chia sẻ ảnh mật

Bài toán chia sẻ ảnh mật

Chia ảnh mật I (ảnh xám, mỗi pixel là một số nguyên $\in [0, 255]$) thành n phần $I^1, I^2, \ldots, I^n$ sao cho:

- Với ít nhất là k phần bất kỳ ($k \leq n$) thì sẽ tái tạo được I
- Với ít hơn k phần thì sẽ không biết gì về I

Mở rộng phương pháp Shamir cho bài toán chia sẻ ảnh mật?

- Một cách đơn giản là áp dụng phương pháp của Shamir cho mỗi pixel trong ảnh mật I (giả sử ảnh I có m pixel)
  - Pixel 1 → phần 1, phần 2, …, phần n
  - Pixel 2 → phần 1, phần 2, …, phần n
  - …
  - Pixel m → phần 1, phần 2, …, phần n

  I^1 I^2 I^n
- Mỗi phần I^i có kích thước như thế nào so với ảnh I?
- Nếu dùng trường $\mathbb{Z}_p$ thì chọn p bằng bao nhiêu?
- Nếu muốn mỗi phần tử trong I^i vẫn là số nguyên $\in [0, 255]$ thì chọn p bằng bao nhiêu?

Phương pháp của Thien & Lin(*) cho bài toán chia sẻ ảnh mật

- Cũng dựa trên phương pháp của Shamir
- Mỗi phần I^i có số lượng phần tử nhỏ hơn so với ảnh I
- Các phần tử trong I^i vẫn là số nguyên $\in [0, 255]$
- Thien & Lin đưa 2 phiên bản là lossy và lossless

(*): Thien & Lin, Secret image sharing, 2002

Phiên bản lossy của Thien & Lin

Quá trình phân chia ảnh mật

- Dùng $p = 251$, là số nguyên tố lớn nhất ≤ 255
- Với các pixel có giá trị > 250, đổi giá trị thành 250
- Duyệt các pixel theo một thứ tự nào đó, cứ k pixel sẽ tạo thành một nhóm (không giao với các nhóm khác). Với mỗi nhóm gồm k pixel $p_1, p_2, \dots, p_k$:
  - Tạo đa thức: $f(x) = p_1 + p_2 x + \cdots + p_k x^{k-1}$
  - Từ đa thức trên tạo ra n phần theo phương pháp Shamir

Như vậy, ta có:

Nhóm 1 → phần 1, phần 2, …, phần n
Nhóm 2 → phần 1, phần 2, …, phần n
…
Nhóm ? → phần 1, phần 2, …, phần n

I^1 I^2 I^n

I^i có kích thước như thế nào so với I?
Bằng kích thước của I chia cho k ☺

Phiên bản lossy của Thien & Lin

Quá trình tái tạo ảnh mật

Với n' phần I^i thì chỉ cần dùng k phần I^i bất kỳ để tái tạo ảnh mật

- Lần lượt duyệt k phần tử thứ 1 của k phần I^i, k phần tử thứ 2 của k phần I^i, …

  Với mỗi lần duyệt: tái tạo lại hàm đa thức bậc $k - 1$ từ k phần tử của k phần I^i, k hệ số của đa thức này chính là k pixel của ảnh mật
- Sau khi đã có tất cả các pixel của ảnh mật, sắp xếp lại thứ tự của các pixel này dựa trên thứ tự đã duyệt khi phân chia

Phiên bản lossless của Thien & Lin

Quá trình phân chia ảnh mật

Giống như phiên bản lossy, chỉ khác là:

- Không có vụ đổi giá trị của các pixel mà > 250 thành 250
- Thay vì làm trên mảng các pixel của I thì sẽ chuyển mảng các pixel của I thành một mảng khác và sẽ làm trên mảng khác này

  Cách chuyển: duyệt mảng các pixel của I:

  - Nếu giá trị pixel $<$ 250: ghi giá trị pixel vào vị trí tương ứng của mảng khác
  - Nếu giá trị pixel $\geq$ 250: tách giá trị pixel thành 2 giá trị là 250 và phần còn lại, rồi ghi 2 giá trị này vào 2 vị trí tương ứng của mảng khác

    → Số lượng phần tử của mảng khác $\geq$ số lượng phần tử của I

Phiên bản lossless của Thien & Lin

Quá trình tái tạo ảnh mật

- Nếu theo đúng các bước của phiên bản lossy thì cuối cùng ta sẽ ra được mảng khác
- Để ra được mảng các pixel của I thì ta sẽ duyệt các phần tử trong mảng khác này:
  - Nếu phần tử có giá trị $\neq 250$: ghi giá trị này vào vị trí tương ứng trong mảng I
  - Nếu phần tử có giá trị $= 250$: lấy thêm phần tử kế trong mảng khác, cộng 2 giá trị này lại và ghi kết quả vào vị trí tương ứng trong mảng I