Khoa Công nghệ thông tin

Lớp: TH2004

Môn: Mã hóa thông tin và Ứng dụng

THẢO LUẬN VỀ MÃ HÓA ĐỐI XỨNG

Sinh viên thực hiện:

* MSSV: 0512175, Họ tên: Nguyễn Đăng Khoa

# Mật khẩu mà người dùng nhập vào thường có độ dài bất kỳ. Có (những) cách nào để tạo ra khóa để mã hóa (thường có quy định cụ thể về độ dài) từ mật khẩu mà người dùng đã nhập. Với mỗi giải pháp mà Anh/Chị đề nghị, hãy tự đánh giá ưu điểm và hạn chế (nếu có).

**Giải pháp:** Có 2 bước:

* Bước 1: Gọi chiều dài của khóa là n, chiều dài của mật khẩu là m (m <= n). (n/2) ký tự đầu tiên của mật khẩu lần lượt được đưa vào các vị trí chẵn: 0, 2, 4, …, n (nếu n chẵn, hoặc n – 1 nếu n lẻ) của khóa. m – (n/2) ký tự còn lại của mật khẩu sẽ được phân phối vào ký tự lẻ: 1, 3, 5, … của khóa. Nếu m < n thì trong khóa sẽ có m – n ký tự chưa được điền vào, tại những ký tự đó sẽ được thêm ký tự \0 vào cho đủ n ký tự của khóa.
* Bước 2: Ta đã có chuỗi mới từ mật khẩu của người dùng. Ta lấy chuỗi này thực hiện xor bit với một chuỗi bit gồm n byte để được chuỗi khóa cuối cùng. Chuỗi bit ở đây chọn ngẫu nhiên (không nên chọn chuỗi bit chỉ toàn 0 hoặc toàn 1 để tăng tính bảo mật hơn).

Ví dụ: Tạo khóa 24 byte, mật khẩu là nguyendangkhoa (14 ký tự).

* **Bước 1:** phân phối 14 ký tự của mật khẩu này vào khóa.
  + Phân phối (n/2) ký tự đầu tiên của mật khẩu vào các ký tự chẵn của khóa.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | n |  | g |  | u |  | y |  | e |  | n |  | d |  | a |  | n |  | g |  | k |  | h |

* + Phân phối m – (n/2) ký tự còn lại của mật khẩu vào các ký tự lẻ của khóa.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| o | n | a | g |  | u |  | y |  | e |  | n |  | d |  | a |  | n |  | g |  | k |  | h |

* + Thêm ký tự \0 vào các vị trí còn bỏ trống của khóa.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| o | n | a | g | \0 | u | \0 | y | \0 | e | \0 | n | \0 | d | \0 | a | \0 | n | \0 | g | \0 | k | \0 | h |

* **Bước 2:** lấy chuỗi vừa có được đem xor với một chuỗi bit kích thước 24 byte. Ví dụ chuỗi bit lấy là: 01010101 01010101 01010101 01010101 01010101 01010101 01010101 01010101. Sau khi xor với chuỗi bit này thì ta có được khóa dùng để mã hóa.

**Nhận xét:**

* **Ưu điểm:**
* Đơn giản, dễ thực hiện, nhanh.
* Các ký tự của mật khẩu được phân bố đều trong khóa. Tránh trường hợp mật khẩu nằm hẳn về một phía của khóa.
* Việc thêm các ký tự \0 mà kô dùng ký tự khác (như ký tự A) vì ký tự này người dùng kô thể gõ vào được. Đảm bảo được sẽ không có 2 mật khẩu mà sau khi phân phối lại giống nhau. Ví dụ nếu chọn ký tự chèn thêm vào là ký tự A thì 2 mật khẩu nguyendangkhoa, nguyendangkhoaAAAAAAAA sau khi biến đổi sẽ cho ra cùng kết quả.
* Cho phép tùy chọn chuỗi bit để đem xor. Mã hóa khóa một lần nữa trước khi dùng.
* **Nhược điểm:**
* Nếu mật khẩu quá ngắn, thì người tấn công dễ dàng biết được chuỗi bit đem xor, bằng cách thử những mật khẩu ngắn. Nếu biết được chuỗi bit, mà người tấn công lại có được chuỗi khóa sau khi mã hóa thì hoàn toàn có thể suy ra được mật khẩu.

# Khi mã hóa 1 tập tin, người dùng có thể chọn thuật toán khác nhau, cũng như chọn các tham số khác nhau trong cùng thuật toán (ví dụ như mode, padding scheme, độ dài khóa, độ dài khối…). Khi giải mã, chúng ta cần biết được chính xác thuật toán với đầy đủ các tham số đã sử dụng. Ngoài phương án cho người sử dụng nhập vào, Anh/Chị hãy đề xuất (những) phương án khác để chương trình tự động xác định được thuật toán và các tham số đã dùng. Với mỗi giải pháp mà Anh/Chị đề nghị, hãy tự đánh giá ưu điểm và hạn chế (nếu có).

**Giải pháp:** dùng 3 byte để biểu diễn các thông tin liên quan đến thuật toán. 3 byte này đặt tại đầu của tập tin. Chi tiết về 3 byte này:

|  |  |
| --- | --- |
| Vị trí (bit) | Mô tả |
| 0 – 2 | Thuật toán   |  |  | | --- | --- | | Giá trị | Thuật toán | | 000 | DES | | 001 | RC2 | | 010 | Triple-DES | | 011 | Rijndael | |
| 3 – 4 | Padding mode   |  |  | | --- | --- | | Giá trị | Padding mode | | 00 | X923 | | 01 | PKCS7 | | 10 | ISO10126 | |
| 5 – 7 | Mode of Operation   |  |  | | --- | --- | | Giá trị | Mode of Operation | | 000 | ECB | | 001 | CBC | | 010 | OFB | | 011 | CFB | | 100 | CTS | |
| 8 – 15 | Độ dài khóa (tính bằng bit) |
| 16 – 23 | Độ dài khối (tính bằng bit) |

* Để tránh người khác đọc được thông tin của 3 byte này, ta tiến hành mã hóa nó, bằng cách: lấy 3 byte ở vị trí (n/2) – 1, n/2, (n/2) + 1 (n là chiều dài đoạn văn bản đã được mã hóa) đem xor với 3 byte thông tin, sẽ được 3 byte mới. 3 byte vừa nhận được sẽ được đặt ở đầu file.

**Nhận xét:**

* **Ưu điểm:**
* Kích thước dùng để lưu trữ thông tin mã hóa nhỏ.
* Cho biết được các thông tin liên quan đến thuật toán mã hóa nhờ đính kèm vào file mã hóa.
* Bảo vệ được thông tin liên quan đến thuật toán mã hóa vì: với cùng thuật toán mã hóa, cùng độ dài khối, khóa, … nhưng nội dung mã hóa khác nhau thì 3 byte sinh ra sẽ khác nhau.
* **Khuyết điểm:**
* Không biết được thông tin liên quan về thuật toán mã hóa có bị thay đổi hay không. Nếu bị thay đổi dẫn đến tình trạng người dùng nhập đúng password, IV nhưng vẫn không giải mã được.