Question 1 (5 points). About asymmetric symmetric encryption

1.1(1 point) Concept.

Mã Hóa Đối Xứng (Symmetric encryption ) là lớp thuật toán các mã hóa trong đó việc mã hóa và giải mã đều dùng chung cho 1 khóa (secret key).

Mã Hóa Bất Đối Xứng (Asymmetric encryption ) Là thuật toán trong đó việc mã hóa và giãi mã dùng hai khóa khác nhau là pulic key(khóa công khai hay khóa công cộng) và private key (khóa riêng ). Nếu dùng public key để mã hóa thì private key sẽ dùng để giải mã và ngược lại

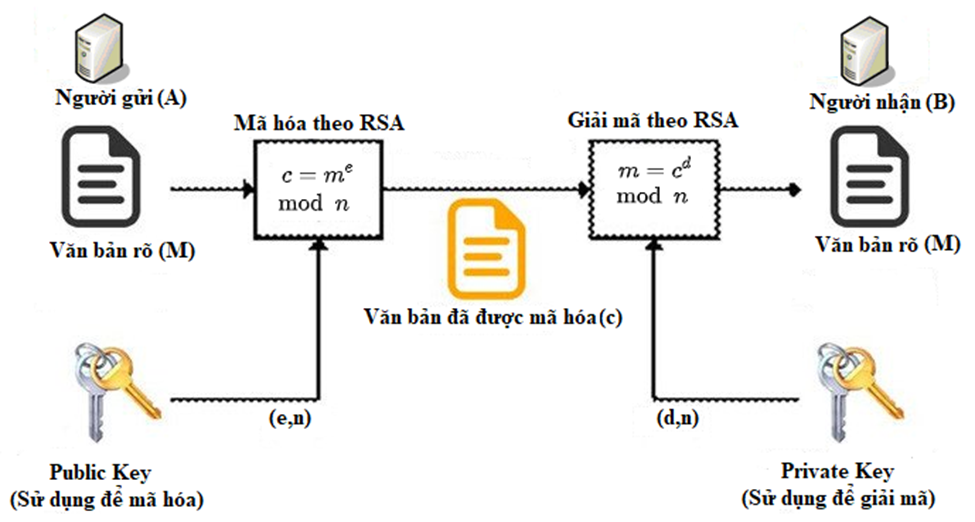
1.2(1 point) Draw an encoding and decoding diagram.

a/Mô Hình mã Hóa đối xứng

A diagram of a key system

Description automatically generated

b/ Mô Hình mã Hóa bất đối xứng



1.3(1 point) Compare.

| **Các thông số so sánh** | **Mã hóa đối xứng** | **Mã hóa bất đối xứng** |
| --- | --- | --- |
| Vài cái tên khác | Mã hóa khóa bí mật, khóa bí mật hoặc mã hóa khóa chung. | Mã hóa khóa công khai |
| Số lượng chìa khóa | Chỉ các khóa riêng được sử dụng. | Sử dụng cả khóa riêng và khóa chung. |
| Thời gian tiêu thụ | Quá trình mã hóa đối xứng diễn ra nhanh chóng. | So với Mã hóa đối xứng, nó chậm hơn. |
| Bảo mật | Mã hóa bất đối xứng kém an toàn hơn. | An toàn hơn quá trình mã hóa đối xứng |
| Các ví dụ | Blowfish, AES, RC4, v.v. | DSA, và RSA, và nhiều hơn nữa. |

* Mã hóa đối xứng sử dụng một bộ khóa duy nhất, trong khi so sánh, đó được gọi là khóa riêng. Mã hóa bất đối xứng sử dụng các bộ khóa khác nhau, khóa chung và khóa riêng, tương ứng được sử dụng bởi người gửi và người nhận.
* Mã hóa đối xứng nhanh hơn quá trình Mã hóa bất đối xứng vì Mã hóa bất đối xứng sử dụng hai bộ khóa khác nhau và Mã hóa đối xứng sử dụng cùng một bộ. Các khóa được sử dụng trong quy trình Mã hóa đối xứng ngắn hơn các khóa được sử dụng trong Mã hóa bất đối xứng.
* Mã hóa bất đối xứng an toàn hơn quy trình Mã hóa đối xứng vì nó ít có khả năng bị tấn công hơn vì chỉ những cá nhân cụ thể mới biết khóa để giải mã dữ liệu hoặc thông tin.
* Mã hóa đối xứng được sử dụng khi một người cần chuyển một lượng thông tin lớn hơn, trong khi Mã hóa bất đối xứng được sử dụng khi thông tin có số lượng nhỏ. Đó là bởi vì Mã hóa bất đối xứng sử dụng một khoảng thời gian lớn hơn để mã hóa và giải mã.
* Mã hóa đối xứng được sử dụng để mã hóa và giải mã dữ liệu trong một khoảng thời gian dài hơn, bắt đầu từ 100 BC. Ngược lại, Mã hóa bất đối xứng xuất hiện muộn hơn nhiều. Do đó, làm cho quá trình Mã hóa đối xứng cũ hơn.

1.4(2 points) Write a program to encrypt and decrypt using asymmetric by C# language.

Question 2 (2 points). Security Vulnerability

1. Concept.

Lỗ hổng bảo mật (security vulnerability) là khuyết điểm trong quá trình lập trình hoặc việc cấu hình sai hệ thống mà qua đó tạo ra sơ hở dẫn dến kẻ tấn công mạng có thể truy cập trực tiếp dữ liệu mà bỏ qua quy trình thông thường.

Việc khai thác lỗ hổng bảo mật còn được gọi là Exploit sẽ giúp các Hacker tận dụng lỗ hổng bảo mật để đạt được lợi ích.

Lỗ hổng bảo mật là một khái niệm rất thường thấy trong lĩnh vực an ninh mạng. Nó được định nghĩa bằng rất nhiều kiểu lý giải khác nhau nhưng chung quy là chỉ về một điểm yếu (kỹ thuật hoặc phi kỹ thuật) của một phần mềm, phần cứng, giao thức, hay một hệ thống thông tin. Người ta thường nhắc đến lỗ hổng bảo mật là lỗi kỹ thuật ở phần mềm, phần cứng chứ không ai nói đó là lỗi ở con người dù lỗi ở con người cũng được tính là lỗ hổng. Ủy ban về Hệ thống An ninh Quốc gia của Hoa Kỳ đã xác định lỗ hổng trong Hướng dẫn CNSS số 4009 ngày 26 tháng 4 năm 2010 thuật ngữ Bảo đảm Thông tin Quốc gia : Vulnerability—Weakness in an information system, system security procedures, internal controls, or implementation that could be exploited by a threat source (tạm dịch Lỗ hổng bảo mật - sự yếu kém trong hệ thống thông tin, quy trình bảo mật hệ thống, kiểm soát nội bộ hoặc việc triển khai có thể bị khai thác bởi nguồn đe dọa).

1. SQL Injection and how to prevent it

SQL Injection là một kỹ thuật lợi dụng những lỗ hổng về câu truy vấn của các ứng dụng. Được thực hiện bằng cách chèn thêm một đoạn SQL để làm sai lệnh đi câu truy vấn ban đầu, từ đó có thể khai thác dữ liệu từ database. SQL injection có thể cho phép những kẻ tấn công thực hiện các thao tác như một người quản trị web, trên cơ sở dữ liệu của ứng dụng.

Ví dụ, trong form đăng nhập, người dùng nhập dữ liệu, trong trường tìm kiếm người dùng nhập văn bản tìm kiếm, trong biểu mẫu lưu dữ liệu, người dùng nhập dữ liệu cần lưu. Tất cả các dữ liệu được chỉ định này đều đi vào cơ sở dữ liệu.

Thay vì nhập dữ liệu đúng, kẻ tấn công lợi dụng lỗ hổng để insert và thực thi các câu lệnh SQL bất hợp pháp để lấy dữ liệu của người dùng… SQL Injection được thực hiện với ngôn ngữ lập trình SQL. SQL (Structured Query Language) được sử dụng để quản lý dữ liệu được lưu trữ trong toàn bộ cơ sở dữ liệu.

Cách giảm thiểu và phòng ngừa SQL Injection

Luôn kiểm tra kỹ các trường nhập dữ liệu và các bạn cần ràng buộc thật kỹ dữ liệu người dùng nhập vào.

Ví dụ:

//Thông thường

$id = $\_GET['id'];

//Ràng buộc

$id = isset($\_GET['id'])?(string)(int)$\_GET['id']:false;

Dùng Regular Expression để loại bỏ đi các ký tự lạ hoặc các ký tự không phải là số.

$id = isset($\_GET['id']) ? $\_GET['id'] : false;

$id = str\_replace('/[^0-9]/', '', $id);

Hoặc dùng các hàm có sẵn để giảm thiểu lỗi. Mỗi khi truy vấn thì mọi người nên sử dụng thêm hàm mysqli\_real\_escape\_string để chuyển đổi một chuỗi thành một query an toàn.

Các tổ chức có thể tập trung vào những bước sau đây để bảo vệ mình khỏi những cuộc tấn công SQL Injection:

– Không bao giờ được tin tưởng những input người dùng nhập vào: Dữ liệu luôn phải được xác thực trước khi sử dụng trong các câu lệnh SQL.

– Các thủ tục được lưu trữ: Những thủ tục này có thể trừu tượng hóa các lệnh SQL và xem xét toàn bộ input như các tham số. Nhờ đó, nó không thể gây ảnh hưởng đến cú pháp lệnh SQL.

– Các lệnh được chuẩn bị sẵn: Điều này bao gồm việc tạo truy vấn SQL như hành động đầu tiên và sau đó xử lý toàn bộ dữ liệu được gửi như những tham số.

– Những cụm từ thông dụng: Những cụm từ này được sử dụng để phát hiện mã độc và loại bỏ nó trước khi câu lệnh SQL được thực hiện.

– Thông báo lỗi đúng: Thông báo lỗi phải tuyệt đối tránh tiết lộ những thông tin/chi tiết nhạy cảm và vị trí xảy ra lỗi trên thông báo lỗi.

– Giới hạn quyền truy cập của người dùng đối với cơ sở dữ liệu: Chỉ những tài khoản có quyền truy cập theo yêu cầu mới được kết nối với cơ sở dữ liệu. Điều này có thể giúp giảm thiểu những lệnh SQL được thực thi tự động trên server.

– Hãy loại bỏ các kí tự meta như ‘”/\; và các kí tự extend như NULL, CR, LF, … trong các string nhận được từ:

input do người dùng đệ trình

các tham số từ URL

các giá trị từ cookie

– Đối với các giá trị numeric, hãy chuyển nó sang integer trước khi query SQL, hoặc dùng ISNUMERIC để chắc chắn nó là một số integer.

– Thay đổi “Startup and run SQL Server” dùng mức low privilege user trong tab SQL Server Security.

Xóa các stored procedure trong database master mà không dùng như:

xp\_cmdshell

xp\_startmail

xp\_sendmail

sp\_makewebtask