Đối tượng hướng tới: thông tin khách hàng

Hướng tới bảo mật thông tin khách hàng ở trong bộ lưu trữ của ngân hàng.

+ Bảo mật từ phía khách hàng:

* Khách hàng đăng nhập vào phần mềm xem thông tin cần có mật khẩu và mã OTP

+ Bảo mật từ phía ngân hàng:

* Thông tin của khách hàng chia làm hai bản: bản cứng và bản mềm
* Khách hàng khi đăng ký online: Thông tin được khách hàng cung cấp sẽ được mã hóa rồi mới chuyển vào bộ lưu trữ của ngân hàng
* Khách hàng khi đăng ký trực tiếp: Thông tin sẽ được ngân hàng nhập vào máy, sau đó mã hóa và chuyển vào bộ lưu trữ của ngân hàng

Các phương pháp bảo mật thông tin khách hàng phổ biến hiện nay

Phương án 1: Mã hóa

Encoding - Mã hóa ký tự là tiến trình chuyển đổi các ký tự Unicode thành biểu diễn nhị phân tương đương của chúng. Khi XML processor đọc một tài liệu XML, nó mã hóa tài liệu phụ thuộc vào kiểu mã hóa. Vì thế, chúng ta cần xác định kiểu mã hóa trong khai báo XML.

**+ Mã hóa HTTPS**: là sự kết hợp giữa HTTP và SSL, TSL sử dụng mã hóa để nhằm tạo nên một rào chắn an ninh, bảo mật khi truyền tải các thông tin.

1, Mã hóa Đối xứng (*symmetric key cryptography*)

Dữ liệu trao đổi giữa 2 bên sẽ được mã hóa và thống nhất cách giải mã, khi đó cho dù attacker có bắt được gói tin những cũng không thể giải mã nếu không biết cách mã hóa

2, Chìa khóa giải mã (key)

Key là chìa khóa giải mã cho dữ liệu đã được mã hóa và gửi qua HTTPS. Mã hóa đối xứng sẽ được bảo mật nếu như chỉ có người gửi và người nhận biết key đã được sử dụng. Những có một vấn đề đó là nếu bên gửi và bên nhận không gặp nhau để thống nhất và thỏa thuận trước khi gửi thì sẽ không biết được key là gì. Còn nếu gửi key trong packet thì khác nào quay lại HTTP làm cho attacker biết được key, giải mã đánh cắp và giả mạo packet. Để giải quyết chúng ta dùng mã khóa không đối xứng.

3, Mã hóa không đối xứng (*asymmetric key cryptography*)

Chúng ta sẽ tạm gọi bên gửi là client và bên nhận là server. Hình thức mã hóa đối xứng sẽ diễn ra như sau:

* Client gửi 1 tin nhắn không chứa dữ liệu cũng như không được mã hóa lên server như 1 request.
* Server gửi lại cho client gói tin có chứa public key và giữ lại private key của server
* Client gửi yêu cầu/ dữ liệu được mã hóa bằng public key mà nó nhận được từ server
* Server nhận được gói tin và giải mã bằng private key.

Vậy là thông tin đã được trao đổi giữa client và server mà không sợ bị attacker đọc trộm hay sửa đổi do attacker không có private key.

4, Chứng nhận thẩm quyền(*certification authority*)

Để tránh attacker đánh cắp gói tin có chứa public key mà server gửi cho client và gửi 1 gói tin chứ public key giả mạo mà attacker có private key. Khi đó client nhận được public key giả mạo nhưng không hề biết mà lại đi mã hóa dữ liệu bằng public key vừa nhận được, sau đó ở bước tiếp theo attacker có thể dễ dàng bắt được gói tin client gửi cho server giải mã bằng private tương ứng để đọc dữ liệu và giử dữ liệu giả mạo đến server sử dụng public key mà nó đã bắt được của server ở bước trước. Khi đó thông tin trao đổi giữa 2 bên sẽ bị lộ và thay đổi.

Để tránh việc này HTTPS dùng một chứng thực khóa công khai (còn gọi là chứng thực số /chứng thực điện tử) là một chứng thực sử dụng chữ kí số để gắn khóa công khai với một thực thể(cá nhân, máy chủ hoặc công ti...). Chứng thực số có thể được sử dụng để xác định một khóa công khai thuộc về ai.

Nhà cung cấp chứng thực số (Certificate Authoirity – CA) phát hành các chứng thực khóa công khai trong đó CA đó chứng nhận khóa công khai nằm trong mỗi chứng thực thuộc về cá nhân, tổ chức, máy chủ hay bất kì thực thể nào ghi cùng trong chứng thực đó. Nhiệm vụ của CA là kiểm tra tính chính xác của thông tin liên quan đến thực thể được cấp chứng thực. Khi người sử dụng tin tưởng vào một CA và họ có thể kiểm tra chữ kí số của CA thì họ cũng có thể tin vào khóa công khai và thực thể ghi trong chứng thực số.

Nhờ có chứng thực khóa công khai mà client có thể biết được public key mà nó nhận được có đúng là public key do server gửi không.

5, Kết hợp giữa Mã hóa đối xứng và Mã hóa không đối xứng

Thay vì mỗi lần trao đổi client và server lại phải thực hiện các bước trao đổi khóa và xác thực khóa, thì sẽ áp dụng cả mã hóa đối xứng và mã hóa không đối xứng bằng cách chỉ trao đổi và xác thực khóa ở giai đoạn đầu mới thiết lập kết nối, sau khi đã trao đổi thành công, client sẽ lưu lại khóa và thực hiện cho các lần sau mà không cần phải trao đổi khóa lại.

Vậy bây giờ ta tạm có 1 quá trình khá an toàn để trao đổi dữ liệu giữa client và server.

Phương án 2: chữ ký số

**Có thể chia chữ ký số thành hai phần chính sau:**

+ Phần cứng – giống một chiếc USB (được gọi là USB token) và được bảo mật bằng mật khẩu hay còn gọi là mã PIN;

+ Chứng thư số là phần không thể tách rời của chữ ký số, chứa tất cả dữ liệu đã được mã hóa của doanh nghiệp.

Chữ ký số **tuyệt đối an toàn** trong quá trình sử dụng. Bởi một trong những đặc điểm của chữ ký số đó là tính bảo mật cao. Khi sử dụng, mỗi người dùng bắt buộc phải có một cặp khóa, bao gồm khóa công khai và khóa bí mật.

Khóa bí mật là loại khóa được dùng để tạo ra chữ ký số. Khóa công khai sẽ được sử dụng với mục đích kiểm tra chữ ký số được tạo bởi khóa bí mật tương ứng.

Vì vậy trong quá trình thực hiện ký số, mọi thông tin trong văn bản/tài liệu sẽ được bảo mật tuyệt đối. Đồng thời, văn bản/tài liệu này chỉ có thể được mở bởi một người duy nhất là người nhận được văn bản/tài liệu đã có ký số.

Chữ ký số thường được sử dụng cho việc xác thực người gửi ằng việc áp dụng mã hóa khóa công khai lại. Để tạo ra một chữ ký số, người gửi mã hóa một thông điệp cới chìa khóa riêng của mình. Trong trường hợp này bất cứ người nhận nào đó có chìa khóa công khai của họ đều có thể đọc nó, song người nhận có thể tin rằng người gửi thực sự là tác giả của thông điệp. Một chữ ký số thường được gắn kèm với thông điệp được gửi, cũng giống như chữ ký viết tay. Tính chân thực và xác nhận được đảm bảo bằng việc sử dụng chữ ký số