**MỤC LỤC**

[**MỞ ĐẦU** 4](#_Toc292701091)

[**I.** **Tổng quan về Web Service** 5](#_Toc292701092)

[***1.*** ***Khái niệm cơ bản*** 5](#_Toc292701093)

[***2.*** ***Các thành phần cơ bản của Web Service*** 5](#_Toc292701094)

[*2.1.* *Security* 6](#_Toc292701095)

[*2.2.* *Reliable Messaging* 7](#_Toc292701096)

[*2.3.* *Transactions* 9](#_Toc292701097)

[*2.4.* *Message and XML* 9](#_Toc292701098)

[***3.*** ***Các phương thức hoạt động của Web Service*** 11](#_Toc292701099)

[*3.1.* *Remote Procedure Calls Model* 11](#_Toc292701100)

[*3.2.* *Representational State Transfer (REST) Model* 12](#_Toc292701101)

[*3.3.* *Message Oriented Model* 12](#_Toc292701102)

[*3.4.* *Service Oriented Model* 13](#_Toc292701103)

[*3.5.* *Resource Oriented Model* 14](#_Toc292701104)

[*3.6.* *Policy Model* 15](#_Toc292701105)

[**II.** **Bảo mật trong Web Service** 16](#_Toc292701106)

[***1.*** ***Tổng quan về chính sách bảo mật*** 16](#_Toc292701107)

[***2.*** ***Mục tiêu*** 17](#_Toc292701108)

[*2.1.* *Cơ chế xác thực (Authentication Mechanisms)* 17](#_Toc292701109)

[*2.2.* *Cơ chế phân quyền (Authorization)* 17](#_Toc292701110)

[*2.3.* *Toàn vẹn dữ liệu và an toàn dữ liệu (Data Integrity and Data Confidentiality)* 18](#_Toc292701111)

[*2.4.* *Toàn vẹn giao dịch và giao tiếp (Integrity of Transaction and Communications)* 18](#_Toc292701112)

[*2.5.* *Loại bỏ từ chối (Non-Repudiation)* 18](#_Toc292701113)

[*2.6.* *An toàn với thông điệp (End-to-End Integrity and Confidentiality of Messages)* 18](#_Toc292701114)

[*2.7.* *Kiểm soát vết giao dịch (Audit Trails)* 18](#_Toc292701115)

[*2.8.* *Các chính sách bảo mật cho các thực thi phân tán (Distributed Enforcement of Security Policies)* 19](#_Toc292701116)

[***3.*** ***Các phương thức tấn công*** 19](#_Toc292701117)

[*3.1.* *Message Alteration (Thay đổi nội dung thông điệp)* 19](#_Toc292701118)

[*3.2.* *Confidentiality (Bảo mật giả)* 20](#_Toc292701119)

[*3.3.* *Man-in-the-middle* 20](#_Toc292701120)

[*3.4.* *Spoofing (Giả mạo)* 21](#_Toc292701121)

[*3.5.* *Denial of Service (Từ chối dịch vụ)* 21](#_Toc292701122)

[*3.6.* *Replay Attacks (Tấn công phát lại)* 23](#_Toc292701123)

[***4.*** ***Các giải pháp bảo mật*** 24](#_Toc292701124)

[*4.1.* *Authentication (Cơ chế xác thực)* 24](#_Toc292701125)

[*4.2.* *Encryption (Mã hóa dữ liệu)* 29](#_Toc292701126)

[*4.3.* *Digital Signature (Chữ kí số)* 30](#_Toc292701127)

[4.3.1. Khái niệm cơ bản 30](#_Toc292701128)

[***5.*** ***Giới thiệu chứng chỉ bảo mật X509*** 34](#_Toc292701129)

[*5.1.* *Tổng quan* 34](#_Toc292701130)

[*5.2.* *Triển khai thực tế* 35](#_Toc292701131)

[**III.** **Lập trình kiểm thử** 39](#_Toc292701132)

[1. ***Đặt vấn đề*** 39](#_Toc292701133)

[***2.*** ***Mục tiêu*** 39](#_Toc292701134)

[***3.*** ***Thực nghiệm*** 39](#_Toc292701135)

[*3.1.* *Phân tích yêu cầu* 39](#_Toc292701136)

[*3.2.* *Môi trường* 40](#_Toc292701137)

[*3.3.* *Chức năng* 40](#_Toc292701138)

[*3.4.* *Các yêu cầu về bảo mật* 40](#_Toc292701139)

[*3.5.* *Các hình thức áp dụng* 40](#_Toc292701140)

[*3.6.* *Kịch bản áp dụng bảo mật* 41](#_Toc292701141)

[*3.7.* *Các bước triển khai* 42](#_Toc292701142)

[***4.*** ***Đánh giá và kết luận*** 47](#_Toc292701143)

[*4.1.* *Đánh giá* 47](#_Toc292701144)

[*4.2.* *Kết luận* 48](#_Toc292701145)

[**KẾT LUẬN** 49](#_Toc292701146)

[**TÀI LIỆU THAM KHẢO** 50](#_Toc292701147)

# MỞ ĐẦU

Từ lâu nay, khái niệm về an toàn bảo mật thông tin luôn gắn liền với lĩnh vực công nghệ thông tin. Đó là một vấn đề hết sức quan trọng và không thể thiếu. Ngày nay, khi mạng toàn cầu đang phát triển với một tốc độ nhanh chóng, với một phạm vi rộng lớn và lượng thông tin được chia sẻ trên mạng tăng lên với tốc độ chóng mặt không ngừng, vấn đề về bảo mật thông tin càng trở thành một yêu cầu hết sức bức thiết. Thật vậy, tất cả những người sử dụng mạng Internet đều có thể chia sẻ thông tin của mình với bất kì ai và với rất nhiều phương thức khác nhau. Việc lợi dụng những sơ hở, lỗ hổng trên mạng để thực hiện đánh cắp dữ liệu, thông tin cá nhân hay đánh cắp thông tin nhằm những mục đích đen sẽ gây hậu quả không thể lường trước. Những hệ quả đó không chỉ ảnh hưởng tới một vài người dùng cá nhân mà còn ảnh hưởng tới các tổ chức, các cơ quan Chính phủ và toàn bộ cộng đồng.

Một đặc trưng quan trọng trên mạng Internet toàn cầu hiện nay đó các dịch vụ trao đổi thông tin trực tuyến, được gọi là Web Service. Đây là phương thức truyển tải dữ liệu, thông tin rất phổ biến trên mạng dành cho bất kì người sử dụng nào. Tuy nhiên, việc đảm bảo an toàn cho thông tin trên web service hiện nay vẫn là một vấn đề hết sức nóng bỏng khi hàng loạt vụ tấn công, xâm nhập vào các trung tâm dữ liệu hay các trường hợp đánh cắp, loan truyền thông tin không được phép đã thực sự gây ra nhiều trấn động trong thế giới mạng. Các cơ quan an ninh thông tin, các tổ chức quốc tế và rất nhiều cá nhân đã và đang nỗ lực trong việc xây dựng nên các cơ chế bảo mật tốt nhất, các giải pháp phòng chống nguy cơ tấn công trên web service. Đó là nội dung của bài tiểu luận mà nhóm sinh viên xin trình bày: **“Bảo mật trong Web Service”**

Trong suốt quá trình thực hiện bài tiểu luận này, nhóm đã rất nỗ lực tìm hiểu kiến thức, tra cứu thông tin nhưng chắc chắn vẫn còn nhiều điểm thiếu sót cần bổ sung, cải thiện trong tương lai. Nhóm xin chân thành cảm ơn PGS.TS. Nguyễn Linh Giang đã tận tình hướng dẫn, giúp đỡ thực hiện tốt bài tiểu luận này.

*Nhóm sinh viên*

1. **Tổng quan về Web Service**
2. ***Khái niệm cơ bản***

Web Service được tổ chức W3C định nghĩa như một phương thức giao tiếp giữa hai ứng dụng qua môi trường mạng. Web Service như một hệ thống phần mềm được thiết kế để hỗ trợ kết nối tương thức máy-máy (machine-to-machine) thông qua mạng. Các web service có một giao diện tương ứng cho các máy thành phần trong quá trình kết nối (được gọi là WSDL – Web Service Description Language). Các hệ thống khác tương tác với Web Service với một cách thức được quy định trong các thông điệp SOAP (Simple Object Access Protocol), chẳng hạn như với giao thức truyền thông dữ liệu siêu văn bản HTTP thì sử dụng các tệp XML (eXtended Mark-up Language) để thực hiện giao tiếp.  
Web Service được cấu thành bởi 2 thành phần cơ bản nhất đó là:

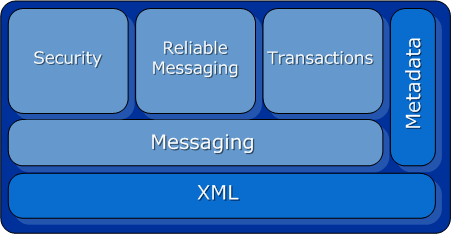
* Agents: là các thành phần phần mềm hay phần cứng cụ thể có chức năng gửi và nhận các thông điệp trên mạng.
* Services: có đặc trưng bao gồm một tập các chức năng chung, cơ bản cần thiết được cung cấp cho các thành phần agents có thể thực hiện qua trình trao đổi thông tin.

Các đối tượng thực hiện việc trao đổi thông tin trong Web Service:

* Requesters: được định nghĩa như những cá nhân hay tổ chức (request entity - đối tượng) hoạt động trên mạng sử dụng các dịnh vụ của Web Service cung cấp. Các đối tượng yêu cầu này sẽ sử dụng requester agent để chuyển vận các thông điệp tới các thực thể cung cấp hồi đáp (provider agent).
* Providers: được định nghĩa như những đối tượng cung cấp các dịch vụ thích hợp để hồi đáp lại phù hợp với các yêu cầu của requester entity.

1. ***Các thành phần cơ bản của Web Service***

Web Service được tổ chức bao gồm các giao thức có thể tương tác qua lại với nhau, gắn kết với nhau như Security, Reliable Messaging và Transactions. Nền tảng của Web Service được xây dựng trên chuẩn giao tiếp của XML và SOAP.



Cơ cấu tổ chức các thành phần chính của Web Service

* 1. *Security*

Bảo mật trong Web Service được Microsoft tổ chức chiến lược bảo mật theo địa chỉ bên trong môi trường Web Service. Đó là một mô hình bảo mật toàn diện trong Web Service cho phép hỗ trợ, tích hợp và thống nhất với một số mô hình, cơ chế hay các công nghệ bảo mật khác, cụ thể, cho phép tương tác rộng giữa nhiều hệ thống trên nền tảng và ngôn ngữ.

Một số định danh trong cơ chế bảo mật Web Service Security

* SOAP Message Security: cung cấp chuẩn chất lượng bảo mật thông qua việc tích hợp thông điệp, bảo mật gửi/nhận thông điệp và xác thực thông điệp trên mạng. Để thực hiện các mục đích xác thực trên thông điệp này, Web Service cung cấp phương thức mã hóa nhị phân các khóa bảo mật như chứng chỉ X.509, Kerberos...
* Web Service Trust: xác định thành phần mở rộng được xây dựng trên Web Security để yêu cầu và xác cung cấp các khóa bảo mật nhằm quản lí các liên kết/quan hệ tin cậy. Ta có thể thấy Web Service Trust được áp dụng trên các trình duyệt web khi truy cập vào các địa chỉ web theo giao thức https://
* Secure Conversation: cung cấp các phương thức giao tiếp truyền thông trên mạng được bảo mật. Chẳng hạn, khi định nghĩa một cơ chế thành lập và chia sẻ thông tin an toàn, ta sẽ cần gửi kèm theo các khóa phiên để đảm bảo việc giao dịch an toàn giữa các bên.
  1. *Reliable Messaging*

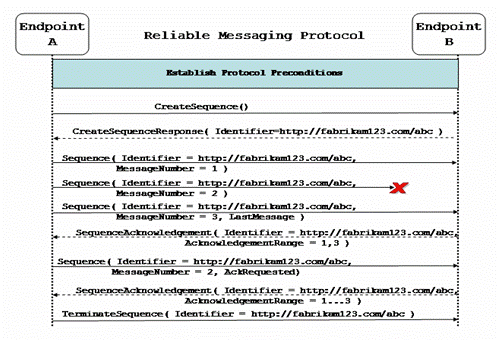
Đây là khái niệm về một giao thức cho phép các thông được chuyển vận đáng tin cậy giữa các ứng dụng phân tán trong các thành phần, các hệ thống phần mềm hay mạng. Giao thức này xác định các cơ chế độc lập, cho phép triển khai áp dụng ở các công nghệ mạng khác nhau. SOAP đã được tích hợp trong giao thức này. Cơ chế chuyển vận thông điệp tin cậy này được áp dụng nhiều trong các ứng dụng phân tán có nhiều tương tranh, đoạn găng.

1. Cơ chế chuyển vận thông điệp tin cậy

* Sequences (tuần tự): các thông điệp được gửi từ địa chỉ nguồn tới địa chỉ đích được giới hạn bằng việc sử dụng dãy tuần tự. Mỗi một thành phần trong trình tự này được gáp một định danh duy nhất (URI – Unique Identifier). Việc tổ chức tuần tự này không tuân theo một thứ tự xử lí nào.
* Message Numbers (số lượng thông điệp): mỗi một thông điệp được gửi nằm trong hàng đợi đó sẽ được gán một mã định danh duy nhất. Giá trị này tăng dần từng đơn vị số nguyên, bắt đầu từ 1 và tăng lên 1 đơn vị tương ứng với mỗi thông điệp. Việc tổ chức đánh số định danh như vậy nhằm tránh hiện tượng trùng lặp các thông điệp và thuận tiện cho việc xử lí sau này.
* Acknowledgments (phản hồi kết quả): đây là phản hồi được gửi trả về khi một thông điệp được chuyển thành công tới địa chỉ đích. Một số phản hồi trong quá trình chuyển vận thông điệp như: xử lí thành công, báo cáo lỗi, yêu cầu giao thức...
* Message persistence (Durability): đây là tính nhất quán của phương thức chuyển vận thông điệp. Như trên ta nói, việc tổ chức tuần tự cho các thông điệp được gửi đi không phải là để xử lí mà chỉ có tác dụng tổ chức, bởi lẽ, với mỗi một thông điệp được gửi đi, tại địa chỉ đích sẽ lưu trữ thông điệp đó cho tới khi thông điệp được xử lí, nhằm đảm bảo rằng mỗi thông điệp đều được xử lí bất kể có thành công hay không, nhất quán thông tin giữa bên gửi và bên nhận.

1. Quá trình tương tác

Quá trình này được chia thành các pha thực hiện. Mỗi pha sẽ thực hiện những chức năng riêng và hoàn thành từng khâu trong quá trình chuyển vận thông điệp giữa hai bên giao dịch.



Sơ đồ mô tả quá trình tương tác giữa hai bên giao dịch

* Phase 1 - Establishing preconditions: trước khi hàng đợi tin cậy được khởi tạo để lưu thông tin về các thông điệp, những điều kiện tiên quyết cần có của giao thức sẽ được thực thi. Các điều kiện này bao gồm những tác vụ như các chính sách chuyển vận, cấu hình điểm cuối, cơ chế an toàn... Các chính sách chuyển vận yêu cầu việc khai báo các thông tin, đặc điểm giữa các bên (endpoint) để áp dụng các yêu cầu an toàn thích hợp.
* Phase 2 – Sequence Creation: giao thức sẽ định nghĩa ra 2 phương thức để khởi tạo định danh cho hàng đợi tuần tự: khởi tạo nguồn và khởi tạo cuối. Khi yêu cầu thực hiện khởi tạo thông tin điểm cuối, bên nguồn sẽ yêu cầu khởi tạo một định danh mới trong hàng đợi tuần tự với một thông điệp CreateSequence; khi đó, bên nhận sẽ hồi đáp lại bên nguồn tương ứng với thông điệp đó bằng một thông điệp CreateSequenceResponse chứa một định danh mới (hoặc lỗi).
* Phase 3 – Communication:

+ Khi bên gửi bứt đầu gửi đi một thông điệp thì trong hàng đợi sẽ xác nhận một mã định danh MessageNumber = 1.

+ Bên nhận sẽ xác định một thông điệp hồi đáp với thông điệp số 1 vừa nhận được từ bên gửi.

+ Bên gửi sẽ cập nhật hồi đáp từ bên nhận để biết được tình trạng thông điệp đã được chuyển thành công hay chưa.

* Phase 4 – Sequence Termination: một xác nhận sẽ được thực hiện nếu bên nhận đã tiếp nhận được thông điệp hoàn toàn và gửi trả lại một hồi đáp thành công. Sau đó, quá trình thực hiện chuyển vận và gửi/đáp thông điệp giữa hai bên sẽ được tiếp tục.
  1. *Transactions*

Đây là khái niệm mô tả các loại/hình thức phối hợp, điều phối trong việc thực hiện các tác vụ trên Web Service, bao gồm 2 loại: Atomic Transaction (AT) và Business Activity (BA). Các nhà phát triển có thể sử dụng đồng thời cả 2 loại trên khi xây dựng các ứng dụng đòi hỏi ràng buộc giữa các tác vụ phân tán.

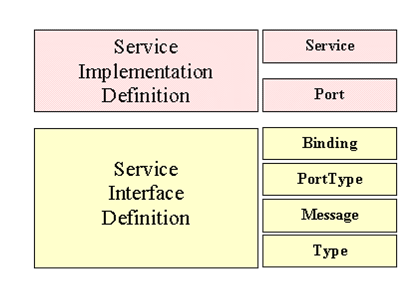
* 1. *Message and XML*

SOAP là một giao thức linh hoạt trong việc chuyển vận thông tin trong các môi trường phân tán. Một giao thức dựa XML gồm 3 thành phần chính là: khung chính cung cấp thông tin về nền tảng được sử dụng trong đó các thông điệp được chuyển vận và xử lí như thế nào; một tập các luật mã hóa cho việc xác định các kiểu dữ liệu làm việc trong ứng dụng; một chuẩn thể hiện các phương thức gọi và hồi đáp. SOAP có thể được sử dụng để kết nối nhiều giao thức khác nhau.

Ý tưởng thực hiện việc chuyển vận thông điệp thông qua nền tảng XML như sau: mỗi một file XML sẽ có các tag lưu lại toàn bộ vết, các thông tin liên quan tới các phiên làm việc (giao dịch) giữa các bên gửi/nhận và bản thân thông điệp trong đó. Việc tổ chức theo XML và SOAP nhằm tận dụng sự linh động, uyển chuyển trong phương thức giao tiếp của SOAP và các thức lưu trữ thông tin hiệu quả, đơn giản của XML.

Dựa trên ý tưởng đó, các nhà tổ chức đã đưa ra một chuẩn mới để thực hiện các giao dịch trên Web Service, kết hợp ưu thế của SOAP và XML, đó là WSDL. WSDL là viết tắt của Web services Description Language. WSDL là tài liệu viết bằng ngôn ngữ XML. Tài liệu này dùng để mô tả dịch vụ web. Khi một ai đó muốn sử dụng dịch vụ của bạn, họ sẽ yêu cầu một tập tin WSDL để tìm ra vị trí của dịch vụ, các lời gọi hàm và làm sao truy cập chúng. Sau đó, họ sử dụng thông tin trong tập tin WSDL để tạo thành một yêu cầu SOAP. WSDL cũng mô tả cách các thông điệp được mã hóa và chỉ ra những gì các giao thức dịch vụ hỗ trợ. WSDL cung cấp một mô tả toàn diện về các dịch vụ của bạn.Nó mô tả những gì mà dịch vụ làm (hoạt động(operations) của nó và các định dạng thông điệp(messages) của nó), làm thế nào để tương tác với nó (các ràng buộc(bindings) và các giao thức của nó), và nơi tìm nó (endpoint URL).Nếu bạn đang cung cấp một dịch vụ, bạn nên luôn luôn cung cấp một WSDL mô tả nó.Nếu bạn là người sử dụng dịch vụ, bạn sử dụng WSDL để xây dựng ứng dụng của bạn.

Cấu trúc một tài liệu WSDL được tổ chức như sau:



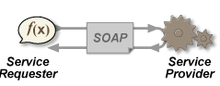
<definitions>  
  
<types>  
  definition of types........  
</types>  
  
<message>  
  definition of a message....  
</message>  
  
<portType>  
  definition of a port.......  
</portType>  
  
<binding>  
  definition of a binding....  
</binding>  
  
</definitions>

Các hình thức tương tác trong WSDL:

|  |  |
| --- | --- |
| One-way | Phương thức có thể nhận 1 thông điệp nhưng không trả về một response. |
| Request-response | Phương thức có thể nhận 1 request và trả về một response. |
| Solicit-response | Gởi 1 request và chờ đợi reponse |
| Notification | Gởi 1 thông điệp nhưng không chờ 1 reponse |

1. ***Các phương thức hoạt động của Web Service***
   1. *Remote Procedure Calls Model*

RPC Web Service có giao diện gọi các chức năng/hàm phân tán rất quen thuộc với các nhà phát triển. Các thành phần cơ bản của RPC Web Service chủ yếu trên cấu trúc tệp WSDL. Đôi khi việc sử dụng RPC cũng gặp một số trục trặc liên quan tới việc liên kết không tốt các thành phần găng của hệ thống bởi giải pháp này thường áp dụng trực tiếp việc ánh xạ giữa các dịch vụ với các lời gọi phương thức hay các hàm chức năng. Một số phương pháp tiếp cận sử dụng ý tưởng này là Object Management Group’s (OMG), Common Object Request Broker Architecture (COBRA), Microsoft’s Distributed Component Object Model (DCOM) hoặc Sun Microsystemt’s Java/Remote Method Invocation (RMI).



* 1. *Representational State Transfer (REST) Model*

REST mô tả một phương thức giao tiếp trên mạng sử dụng HTTP hoặc các giao thức tương tự, bao gồm cả những tập giao diện, hoạt động chuẩn (như GET, POST, PUT, DELETE). Ở đây, giải pháp này tập trung vào việc liên kết các tài nguyên có trạng thái hơn cả thông điệp và chức năng.

Một kiến trúc dựa trên REST (RESTful) có thể sử dụng cấu trúc WSDL để mô tả thông điệp SOAP thông qua giao thức HTTP. WSDL phiên bản 2.0 hỗ trợ việc sử dụng tất cả các phương thức HTTP Request nên việc triển khai sẽ hiệu quả hơn kiến trúc RESTful Web Service.

* 1. *Message Oriented Model*

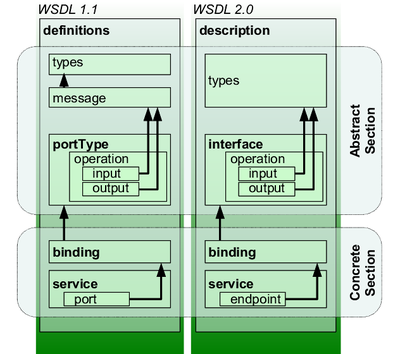
Kiến trúc Message Oriented Model tập trung vào các khía cạnh của kiến trúc hướng thông điệp và xử lí. Đặc biệt, ở kiến trúc này, người ta không tập trung vào các ngữ nghĩa đặc trưng bên trong nội dung của thông điệp hoặc các mối quan hệ tới các thông điệp khác. Tuy nhiên, MOM tập trung vào cấu trúc của thông điệp, các mối quan hệ giữa người gửi thông điệp với người nhận thông điệp và phương thức chuyển vận các thông điệp đó.



Mô hình kiến trúc Message Oriented Model

* 1. *Service Oriented Model*

Web Service có thể được phát triển dựa trên kiến trúc hướng dịch vụ Service-oriented Architecture (SOA), các thành phần chính sẽ giao tiếp với nhau thông qua các thông điệp. SOA Web Service được hỗ trợ bởi rất nhiều nhà phát triển phần mềm. Không giống như RPC Web Service, liên kết lỏng lẻo (loosely coupling) được tổ chức hiệu quả hơn vì kiến trúc này tập trung vào các giao dịch được WSDL hỗ trợ. Cơ chế tổ chức Middleware sử dụng trong kiến trúc này sẽ kết nối, xử lí các thông điệp. Một ví dụ về kiến trúc hướng thông điệp đó là ESB với phiên bản Mule.



Sơ đồ tổ chức REST

* 1. *Resource Oriented Model*

Kiến trúc này tập trung vào các vấn đề liên quan tới tài nguyên của hệ thống. Tài nguyên ở đây bao gồm các thành phần mà Web và Web Service sử dụng. ROM tập trung vào các đặc trưng của tài nguyên, các ràng buộc về quyền hạn sử dụng tài nguyên trên Web Service, các cơ chế xác thực với tài nguyên...



Mô hình kiến trúc Resource Oriented Model

* 1. *Policy Model*

Policy Model tập trung tới các cơ chế, chính sách trong Web Service như các thành phần mở rộng, bảo mật, quyền hạn sử dụng hay chất lượng dịch vụ. Tổng quan của kiến trúc này mô tả chi tiết về các chính sách bảo mật, các ràng buộc của các thành phần bên trong hệ thống và các quan hệ giữa các yếu tố khác đối với toàn bộ kiến trúc.



Sơ đồ mô phỏng mô hình Policy trong Web Service

1. **Bảo mật trong Web Service**
2. ***Tổng quan về chính sách bảo mật***

Dựa trên những quan điểm thiết kế kiến trúc cho Web Service như đã nêu, có 3 nguyên tắc chung được xây dựng cho việc bảo mật Web Service, đó là: tài nguyên phải được bảo mật; các cơ chế được thực hiện trên các tài nguyên này phải được bảo mật; các chính sách được các thiết bị thực thi phải có ràng buộc liên quan tới các tài nguyên đó.

Các chính sách có thể được phân chia thành 2 loại chính: các chính sách cho phép và các chính sách bắt buộc. Một chính sách cho phép tập trung vào các tác vụ và các truy cập được cấp phép cho các thực thể trong quá trình tương tác trên Web Service. Với chính sách bắt buộc thì tập trung vào các tác vụ và các truy cập bắt buộc cần có (điều kiện tiên quyết) đối với các thực thể. Các yếu tố đó có quan hệ chặt chẽ với nhau và phụ thuộc vào nhau; tuy nhiên, không có ràng buộc giữa việc bắt buộc thực hiện các yêu cầu nào mà không được cấp quyền, có nghĩa là nếu một yêu cầu bạn không có quyền thì cũng không cần thiết yêu cầu bạn bắt buộc phải có. Thông thường, người ta xây dựng các chính sách riêng cụ thể bằng việc tổng hớp cả những yêu cầu trên chính sách bắt buộc lẫn chính sách cho phép.

Mỗi một kiến trúc Web Service được thiết kế ra đều chủ ý xây dựng những cơ chế luật, phân quyền người dùng dựa trên nhu cầu và đặc thù của từng bên gửi/nhận (requester/provider). Chẳng hạn, một giám sát viên của một hệ thống sẽ thông kê toàn bộ những quyền hạn truy nhập hệ thống đối với một đối tượng nhất định; với bất kì yêu cầu truy xuất nào đối với tài nguyên của hệ thống, đối tượng đó đều phải được xác thực thông tin, quyền hạn rõ ràng nằm trong những yêu cầu cho phép và bắt buộc đã được đề xuất như trên.

Hiện nay, các tổ chức quốc tế đã xây dựng nên những cơ chế có mực độ bảo mật trên mạng rõ ràng như Transport Layer Security (SSL/TLS), Virtual Private Networks (VPNs), IPSec (Internet Protocol Security) và Secure Multipurpose Internet Mail Exchange (S/MIME)... Mặc dù các cơ chế đó đã được áp dụng trên Web Service, tuy nhiên, vẫn chưa thể cung cấp đầy đủ được cho tất cả các bối cảnh có thể xảy ra, chẳng hạn cần chi tiết hơn. Nhìn chung, Web Service sử dụng phương thức giao tiếp thông qua thông điệp đã tạo ra rất nhiều những hình thức tương tác phức tạp đòi hỏi việc tổ chức, định tuyến các thông điệp giữa rất nhiều phạm vi.



Bảo mật End-to-End trên Web Service

1. ***Mục tiêu***

Web Service là phương thức giao dịch phổ biến trên mạng hiện nay. Chính vì thế, mong muốn có thể xây dựng một môi trường mà các thông điệp được chuyển vận có được mức độ bảo mật cao nhất.

* 1. *Cơ chế xác thực (Authentication Mechanisms)*

Cơ chế xác thực là cần thiết trong việc xác định danh tính đích thực của người gửi yêu cầu và người đáp trả yêu cầu. Trong một số trường hợp, việc xác thực cần thiết ngay từ khi những người tham gia không cần thiết phải kết nối vào giao dịch. Dựa trên các chính sách bảo mật đã có thì có thể xác thực ra danh tính của người gửi yêu cầu (requesters), người tiếp nhận (receivers)...

Một vài phương thức xác thực phổ biến hiện nay như: yêu cầu mật khẩu, mật khẩu xác thực một lần, xác thực bằng độ mạnh của mật khẩu, đa mật khẩu (gồm nhiều mật khẩu con ghép lại)...Một số giải pháp hiệu quả được áp dụng cho việc xác thực người dùng hiện nay như: Lightweight Directory Access Protocol (LDAP), Remote Authentication Dial-in User Service (RADIUS), Kerberos and Public Key Infrastructure (PKI)...

* 1. *Cơ chế phân quyền (Authorization)*

Cơ chế phân quyền nhằm mục đích điều khiển quyền truy cập tài nguyên của người dùng theo những chiến lược đã định sẵn. Sau khi đã qua giai đoạn xác thực (nêu trên), cơ chế phân quyền sẽ cấp cho người gửi yêu cầu những quyền nhất định để truy xuất vào tài nguyên của hệ thống.

* 1. *Toàn vẹn dữ liệu và an toàn dữ liệu (Data Integrity and Data Confidentiality)*

Kĩ thuật tích hợp dữ liệu đảm bảo thông tin không bị pha tạp hoặc bị chỉnh sửa trong quá trình chuyển vận mà không có sự giám sát. Khi đó, dữ liệu sẽ được đảm bảo rằng chỉ có những đối tượng cụ thể được phép truy xuất. Dữ liệu được mã hóa kĩ thuật Chữ kí số sẽ được sử dụng với mục đích này.

* 1. *Toàn vẹn giao dịch và giao tiếp (Integrity of Transaction and Communications)*

Cơ chế này đảm bảo rằng việc sử lí giao dịch được thực hiện đúng cách và trình tự các xử lí được thực hiện đúng như cơ chế đã được xây dựng sẵn.

* 1. *Loại bỏ từ chối (Non-Repudiation)*

Đây là một dịch vụ bảo mật cho phép các giao dịch chống lại việc từ chối sai được tạo ra bởi một giao dịch khác (can thiệp bất hợp pháp). Dịch vụ này sẽ cung cấp một bên thứ ba để đảm bảo việc thống nhất giữa 2 phía giao dịch.

* 1. *An toàn với thông điệp (End-to-End Integrity and Confidentiality of Messages)*

Đảm bảo an toàn với sự hiện diện của các thành phần trung gian tham gia trong quá trình giao dịch.

* 1. *Kiểm soát vết giao dịch (Audit Trails)*

Mục đích của cơ chế này nhằm quản lí, lưu lại toàn bộ vết giao dịch của người dùng và bất cứ thao tác nào. Cơ chế này sẽ được thực hiện bởi một tác tử giám sát luôn bám theo các hoạt động của hệ thống và ghi lại toàn bộ các thông tin trong quá trình giao dịch.

* 1. *Các chính sách bảo mật cho các thực thi phân tán (Distributed Enforcement of Security Policies)*

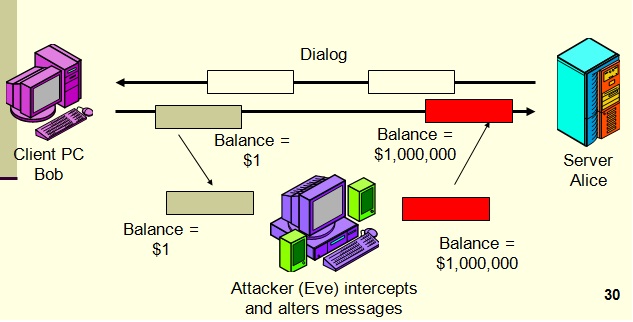
Việc triển khai của dịch vụ này cần xác định ra các chính sách an ninh và bảo đảm việc thực hiện trên nhiều nền tảng khác nhau, ở nhiều trạm khác nhau...

1. ***Các phương thức tấn công***

Hiện nay, các đối tượng tấn công các hệ thống trực tuyến thông qua Web Service ngày càng có nhiều thủ đoạn tinh vi. Một số phương thức chủ yếu thường được áp dụng sẽ được liệt kê sau đây:

* 1. *Message Alteration (Thay đổi nội dung thông điệp)*

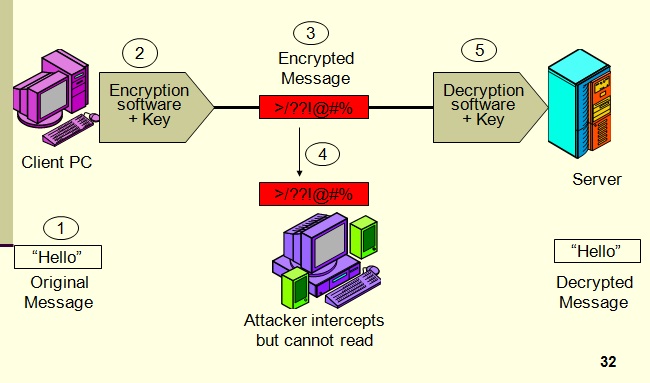
Hình thức tấn công này sẽ chỉnh sửa lại nội dung một số phần hoặc toàn bộ thông điệp như: xóa bỏ một số phần của thông điệp, viết lại nội dung một số phần trong thông điệp, thêm vào một số thông bên ngoài vào thông điệp... Hình thức này chủ yếu thực hiện ở phần đầu hoặc phần thân của thông điệp. Ngoài ra, những kẻ tấn công có thể đính kèm theo thông điệp những nội dung cụ thể do chúng tự dựng nên nhằm thực hiện những mục đích riêng.



Sơ đồ mô phỏng tấn công Message Alteration

**Giải pháp:**

Ở hai phía gửi và nhận thông điệp cần có các khóa và giải thuật để mã hóa và giải mã hóa tương ứng. Khi đó, thông điệp được gửi sẽ được mã hóa an toàn, nếu kẻ tấn công có chặn được gói tin trong quá trình giao dịch nhưng không có được khóa và giải thuật giải mã tương ứng cũng không thể hiểu được nội dung thông điệp.



Giải pháp chống kẻ tấn công thực hiện việc bắt và thay đổi nội dung thông điệp trên đường truyền

* 1. *Confidentiality (Bảo mật giả)*

Với hình thức này, những đối tượng không được xác thực, không được phân quyền sẽ mang trong mình những thông tin xác thực giả mạo để đánh lừa những rào cản bảo mật của hệ thống. Ví dụ: gắn kèm thông tin credit card trong thông điệp gửi đi để giả mạo một người dùng tin cậy...

* 1. *Man-in-the-middle*

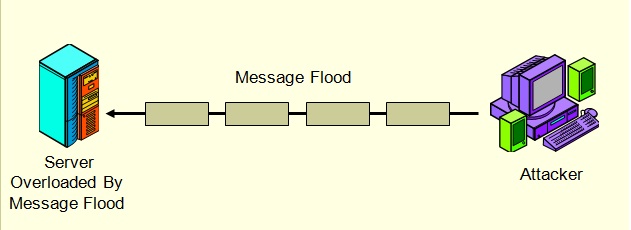
Hình thức này còn được biết đến với tên gọi khác là bucket-brigade. Ở đây, kẻ tấn công sẽ tương tác trực tiếp vào giao dịch SOAP và thu lấy thông điệp đang được gửi giữa requester và receiver; sau đó, chúng sẽ chỉnh sửa hoặc thực hiện các mục đích đen với các thông điệp đó. Cơ chế xác thực sẽ có tác dụng chống lại hình thức tấn công này.

* 1. *Spoofing (Giả mạo)*

Đây là hình thức tấn công phức tạp được thực hiện ngay trong việc khai thác các quan hệ tin cậy. Kẻ tấn công sẽ giả định danh của một người dùng tin cậy nhằm mục đích phá hoại người được nhận thông điệp. Các cơ chế xác thực mạnh mẽ cần triển khai để ngăn chặn hình thức tấn công này.

* 1. *Denial of Service (Từ chối dịch vụ)*

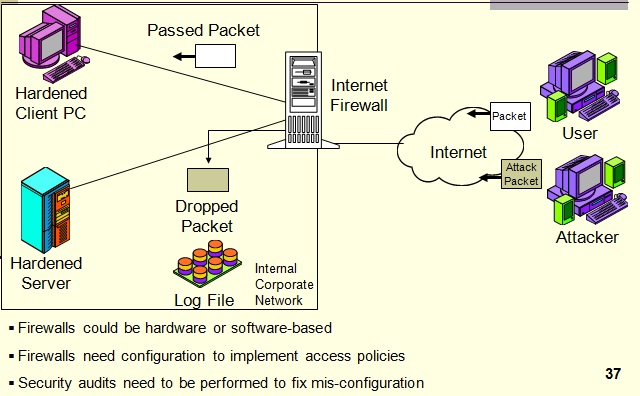
Từ chối dịch vụ là cơ chế tấn công rất phổ biến, nhắm cản trở việc truy cập dịch vụ hợp pháp của người sử dụng (không cho phép việc sử dụng dịch vụ). DoS khai thác điểm yếu trong kiến trúc của hệ thống để tấn công, đó là các lỗ hổng bảo mật. Việc tấn công này sẽ ngăn chặn hoàn toàn việc kết nối tới hệ thống của người dùng. Một hình thức khác của hình thức tấn công này là Distributed Denial of Service (DDoS) nhằm vào các cơ sở dữ liệu, tài nguyên bằng việc tấn công DoS đồng loạt vào trung tâm cơ sở dữ liệu nhằm phá hoại các cơ sở dữ liệu này.



Sơ đồ mô phỏng quá trình tấn công DoS

Giải pháp:

* Sử dụng Firewall:



Sơ đồ mô phỏng phương thức chống tấn công DoS

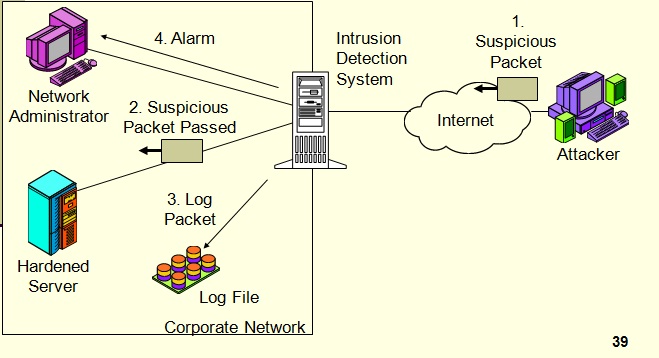
sử dụng Firewall

+ Firewall có thể được xây dựng dựa trên các thiết bị phần cứng hay các phần mềm

+ Firewall cần được cấu hình để triển khai các chính sách truy cập được định sẵn

+ Các thống kê bảo mật cần được thể hiện chi tiết để phục vụ cho việc cấu hình các chính sách bảo mật

* Sử dụng IDS (Intrution Detection System):



Sơ đồ mô phỏng giải pháp chống tấn công DoS sử dụng IDS

+ Thu bắt và lưu các hoạt động giao dịch trên mạng rồi lưu vào các file log (log data files)

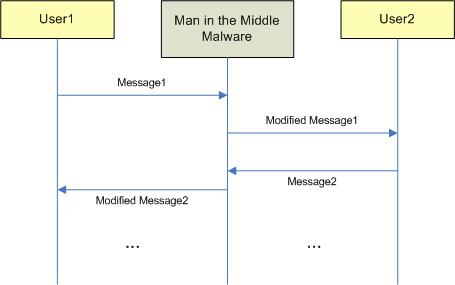
+ Phân tích các hoạt động đã được thu bắt đó

+ Đưa ra các hình thức báo động xâm nhập tương ứng với các kết quả phân tích trên

Cơ chế thực hiện của IDS

* 1. *Replay Attacks (Tấn công phát lại)*

Đây là hình thức mà một kẻ đột nhập sẽ ngăn chặn thông điệp và sau đó sẽ gửi lại tới người nhận.



Sơ đồ mô phỏng quá trình tấn công Man-in-the-middle

1. ***Các giải pháp bảo mật***

Một số cơ chế bảo mật được đề xuất sử dụng có hiệu quả trên Web Service hiện nay như:

* 1. *Authentication (Cơ chế xác thực)*

Đây là giải pháp quan trọng trong việc thực hiện các giao dịch giữa các bên tham gia. Mỗi một thành viên máy khách đưa ra các yêu cầu SOAP để thông báo cho máy chủ server biết về thông tin cần thiết, từ đó đưa ra các giao thức xác thực tương ứng.

Một trong số những giải pháp xác thực được đề xuất là Kerberos. **Kerberos** là một giao thức mật mã dùng để xác thực trong các mạng máy tính hoạt động trên những đường truyền không an toàn. Giao thức Kerberos có khả năng chống lại việc nghe lén hay gửi lại các gói tin cũ và đảm bảo tính toàn vẹn của dữ liệu. Mục tiêu khi thiết kế giao thức này là nhằm vào mô hình máy chủ-máy khách (*client-server*) và đảm bảo nhận thực cho cả hai chiều.

Giao thức được xây dựng dựa trên mật mã hóa khóa đối xứng và cần đến một bên thứ ba mà cả hai phía tham gia giao dịch tin tưởng.

Học viện kỹ thuật Massachusetts (MIT) phát triển Kerberos để bảo vệ các dịch vụ mạng cung cấp bởi dự án Athena. Tên của giao thức được đặt theo tên của con chó ba đầu Cerberus canh gác cổng địa ngục trong thần thoại Hy Lạp. Giao thức đã được phát triển dưới nhiều phiên bản, trong đó các phiên bản từ 1 đến 3 chỉ dùng trong nội bộ MIT.

Các tác giả chính của phiên bản 4, Steve Miller và Clifford Neuman, đã xuất bản giao thức ra công chúng vào cuối thập niên 1980, mặc dù mục đích chính của họ là chỉ phục vụ cho dự án Athena.

Phiên bản 5, do John Kohl và Clifford Neuman thiết kế, xuất hiện trong tài liệu RFC 1510 vào năm 1993 (được thay thế bởi RFC 4120 vào năm 2005) với mục đích sửa các lỗi của phiên bản 4.

MIT đã cung cấp các phiên bản thực hiện Kerberos miễn phí dưới giấy phép tương tự như dùng cho các sản phẩm BSD.

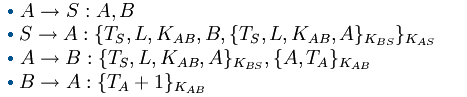
Chính phủ Hoa Kỳ đã cấm xuất khẩu Kerberos vì nó có sử dụng thuật toán DES 56 bit. Tuy nhiên, trước khi chính sách xuất khẩu của Hoa Kỳ thay đổi (2000), đã có phiên bản KTH-KRB viết tại Thụy Điển thực hiện Kerberos 4 được phân phối rộng rãi bên ngoài Hoa Kỳ. Phiên bản này được dựa trên một phiên bản khác có tên là eBones. eBones lại dựa trên một phiên bản được xuất khẩu của MIT thực hiện Kerberos 4 (patch-level 9) gọi là Bones (loại bỏ các hàm mật mã và các lệnh gọi chúng). Eric Young, một lập trình viên người Australia, đã phục hồi lại các lệnh gọi hàm và sử dụng các hàm mật mã trong thư viện của anh ta. Một phiên bản khác thực hiện Kerberos 5, Heimdal, cũng được thực hiện bởi nhóm đã xuất bản KTH-KRB.

Các hệ điều hành Windows 2000, Windows XP và Windows Server 2003 sử dụng một phiên bản thực hiện Kerberos làm phương pháp mặc định để nhận thực. Những bổ sung của Microsoft vào bộ giao thức Kerberos được đề cập trong tài liệu RFC 3244 ("Microsoft Windows 2000 Kerberos Change Password and Set Password Protocols"). Hệ điều hành Mac OS X cũng sử dụng Kerberos trong các phiên bản máy khách và máy chủ của mình.

**Nguyên tắc hoạt động của Kerberos:** Kerberos được thiết kế dựa trên giao thức Needham-Schroeder. Kerberos sử dụng một bên thứ ba tham gia vào quá trình nhận thực gọi là "trung tâm phân phối khóa" (tiếng Anh: key distribution center - KDC). KDC bao gồm hai chức năng: "máy chủ xác thực" (authentication server - AS) và "máy chủ cung cấp vé" (ticket granting server - TGS). "Vé" trong hệ thống Kerberos chính là các chứng thực chứng minh nhân dạng của người sử dụng.

Mỗi người sử dụng (cả máy chủ và máy khách) trong hệ thống chia sẻ một khóa chung với máy chủ Kerberos. Việc sở hữu thông tin về khóa chính là bằng chứng để chứng minh nhân dạng của một người sử dụng. Trong mỗi giao dịch giữa hai người sử dụng trong hệ thống, máy chủ Kerberos sẽ tạo ra một khóa phiên dùng cho phiên giao dịch đó.

Mô tả giao thức: Theo hệ thống ký hiệu giao thức mật mã, Kerberos được mô tả như sau (trong đó Alice (A) sử dụng máy chủ (S) để nhận thực với Bob (B)):



An ninh của giao thức phụ thuộc rất nhiều vào các trường T (đánh dấu thời điểm) và L (thời hạn) của các gói tin. Đây chính là các chỉ thị về tính chất mới của các gói tin và chống lại các tấn công gửi lại các gói tin cũ (xem thêm: Lô gic BAN).

Trong các bản tin ở trên, máy chủ S bao gồm cả dịch vụ nhận thực và sẽ được cung cấp vé. Như vậy, trong gói tin C:\Users\Thangbeomerock\Desktop\a13d728159c84d8e24f4e04fa78e20e5.png , KAB chính là khóa phiên giữa A và B; C:\Users\Thangbeomerock\Desktop\d811ef869fb815d50fa2741cb16a5573.png là vé gửi từ máy khách tới máy chủ; là phần để nhận thực A với B; và C:\Users\Thangbeomerock\Desktop\19d7ebe3c51992069fb5a02acf4ec1be.png để khẳng định lại nhân dạng của B và thông qua đó chấp nhận A. Điều này cần thiết để hai bên nhận dạng lẫn nhau.

Cơ chế hoạt động:

Sau đây là mô tả một phiên giao dịch (giản lược) của Kerberos. Trong đó: AS = Máy chủ chứng thực (*authentication server*), TGS = Máy chủ cấp vé (*ticket granting server*), SS = Máy chủ dịch vụ (*service server*).

Một cách vắn tắt: người sử dụng chứng thực mình với máy chủ chứng thực AS, sau đó chứng minh với máy chủ cấp vé TGS rằng mình đã được chứng thực để nhận vé, cuối cùng chứng minh với máy chủ dịch vụ SS rằng mình đã được chấp thuận để sử dụng dịch vụ.

1. Người sử dụng nhập tên và mật khẩu tại máy tính của mình (máy khách).
2. Phần mềm máy khách thực hiện hàm băm một chiều trên mật khẩu nhận được. Kết quả sẽ được dùng làm khóa bí mật của người sử dụng.
3. Phần mềm máy khách gửi một gói tin (không mật mã hóa) tới máy chủ dịch vụ AS để yêu cầu dịch vụ. Nội dung của gói tin đại ý: "người dùng XYZ muốn sử dụng dịch vụ". Cần chú ý là cả khóa bí mật lẫn mật khẩu đều không được gửi tới AS.
4. AS kiểm tra nhân dạnh của người yêu cầu có nằm trong cơ sở dữ liệu của mình không. Nếu có thì AS gửi 2 gói tin sau tới người sử dụng:
   * Gói tin A: "Khóa phiên TGS/máy khách" được mật mã hóa với khóa bí mật của người sử dụng.
   * Gói tin B: "Vé chấp thuận" (bao gồm chỉ danh người sử dụng (ID), địa chỉ mạng của người sử dụng, thời hạn của vé và "Khóa phiên TGS/máy khách") được mật mã hóa với khóa bí mật của TGS.
5. Khi nhận được 2 gói tin trên, phần mềm máy khách giải mã gói tin A để có khóa phiên với TGS. (Người sử dụng không thể giải mã được gói tin B vì nó được mã hóa với khóa bí mật của TGS). Tại thời điểm này, người dùng có thể nhận thực mình với TGS.
6. Khi yêu cầu dịch vụ, người sử dụng gửi 2 gói tin sau tới TGS:
   * Gói tin C: Bao gồm "Vé chấp thuận" từ gói tin B và chỉ danh (ID) của yêu cầu dịch vụ.
   * Gói tin D: Phần nhận thực (bao gồm chỉ danh người sử dụng và thời điểm yêu cầu), mật mã hóa với "Khóa phiên TGS/máy khách".
7. Khi nhận được 2 gói tin C và D, TGS giải mã D rồi gửi 2 gói tin sau tới người sử dụng:
   * Gói tin E: "Vé" (bao gồm chỉ danh người sử dụng, địa chỉ mạng người sử dụng, thời hạn sử dụng và "Khóa phiên máy chủ/máy khách") mật mã hóa với khóa bí mật của máy chủ cung cấp dịch vụ.
   * Gói tin F: "Khóa phiên máy chủ/máy khách" mật mã hóa với "Khóa phiên TGS/máy khách".
8. Khi nhận được 2 gói tin E và F, người sử dụng đã có đủ thông tin để nhận thực với máy chủ cung cấp dịch vụ SS. Máy khách gửi tới SS 2 gói tin:
   * Gói tin E thu được từ bước trước (trong đó có "Khóa phiên máy chủ/máy khách" mật mã hóa với khóa bí mật của SS).
   * Gói tin G: phần nhận thực mới, bao gồm chỉ danh người sử dụng, thời điểm yêu cầu và được mật mã hóa với "Khóa phiên máy chủ/máy khách".
9. SS giải mã "Vé" bằng khóa bí mật của mình và gửi gói tin sau tới người sử dụng để xác nhận định danh của mình và khẳng định sự đồng ý cung cấp dịch vụ:
   * Gói tin H: Thời điểm trong gói tin yêu cầu dịch vụ cộng thêm 1, mật mã hóa với "Khóa phiên máy chủ/máy khách".
10. Máy khách giải mã gói tin xác nhận và kiểm tra thời gian có được cập nhật chính xác. Nếu đúng thì người sử dụng có thể tin tưởng vào máy chủ SS và bắt đầu gửi yêu cầu sử dụng dịch vụ.
11. Máy chủ cung cấp dịch vụ cho người sử dụng.

**Nhược điểm:**

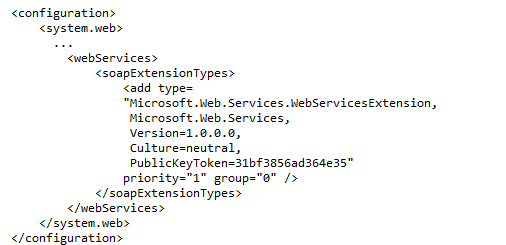
* Tồn tại một điểm yếu: Nếu máy chủ trung tâm ngừng hoạt động thì mọi hoạt động sẽ ngừng lại. Điểm yếu này có thể được hạn chế bằng cách sử dụng nhiều máy chủ Kerberos.
* Giao thức đòi hỏi đồng hồ của tất cả những máy tính liên quan phải được đồng bộ. Nếu không đảm bảo điều này, cơ chế nhận thực giữa trên thời hạn sử dụng sẽ không hoạt động. Thiết lập mặc định đòi hỏi các đồng hồ không được sai lệch quá 10 phút.
* Cơ chế thay đổi mật khẩu không được tiêu chuẩn hóa.
  1. *Encryption (Mã hóa dữ liệu)*

Việc chuyển vận dữ liệu trên mạng dù theo bất kì chuẩn định dạng nào cũng đều tạo cơ hội cho những kẻ xâm nhập truy xuất trái phép vào dữ liệu. Với cơ chế truyền dữ liệu sử dụng SOAP và XML cũng không tránh khỏi tình trạng đó. Bởi thế, người ta sẽ mã hóa các tệp tin gửi trên mạng nhằm đảm bảo dữ liệu và cấu trúc thông điệp gửi đi được bảo vệ tốt nhất. Mã hóa là phương thức xử lí với việc chuyển đổi dữ liệu gốc sang các dạng dữ liệu khác che dấu dữ liệu gốc với các khóa đặc biệt để mã hóa và giải mã. Cơ chế đó không cho phép việc truyền dữ liệu trực tiếp trên Web Service mà đòi hỏi phải mã hóa trước khi truyền, đồng thời bên phía người nhận sẽ phải thiết lập các kênh bảo mật riêng để mã hóa và giải mã tương ứng. Các thuật toán mã hóa được chia làm hai loại chính là đối xứng (shared-secret) và bất đối xứng (public-key)

**Cơ chế mã hóa XML:**

Giao thức mã hóa XML sẽ xác định các phần nằm trong thân của thông điệp SOAP để thực hiện mã hóa. Khi mã hóa XML, thuật toán mã hóa sẽ duyệt lại tệp tin XML và những thông tin trong tệp tin này sẽ được thay thế bởi dữ liệu đã được mã hóa bên trong thành phần EncryptedData. Kết quả của việc mã hóa EncryptedData sẽ được đặt trong thẻ Secuirty header. Nếu có nhiều trường trong tệp tin đó được mã hóa thì mỗi thành phần được mã hóa đó sẽ được ánh xạ tương ứng sang từng thành phần ReferencedData trong ReferenceList.

Ví dụ về mã hóa thông điệp SOAP: với trường Payment được mã hóa thông tin của khách hàng:

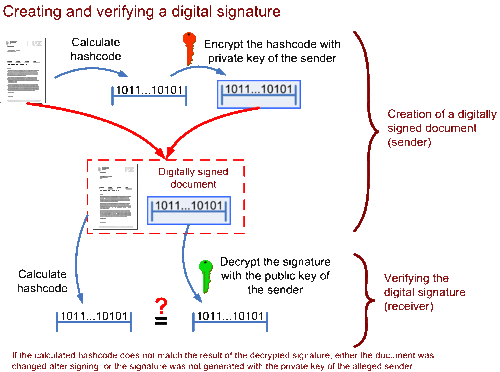


Một trong những thuật toán được sử dụng phổ biến là X.509 Certificates.

* 1. *Digital Signature (Chữ kí số)*
     1. Khái niệm cơ bản

Chữ ký điện tử là thông tin đi kèm theo dữ liệu (văn bản, hình ảnh, video...) nhằm mục đích xác định người chủ của dữ liệu đó.

*Chữ ký số khóa công khai* (hay *hạ tầng khóa công khai*) là mô hình sử dụng các kỹ thuật mật mã để gắn với mỗi người sử dụng một cặp khóa công khai - bí mật và qua đó có thể ký các văn bản điện tử cũng như trao đổi các thông tin mật. Khóa công khai thường được phân phối thông qua chứng thực khóa công khai. Quá trình sử dụng chữ ký số bao gồm 2 quá trình: tạo chữ ký và kiểm tra chữ ký.



Sơ đồ tạo và kiểm tra chữ kí số (sưu tầm)

Cấu trúc Signature Header được xác định trông tệp tin XML sử dụng trong giao thức SOAP như sau :



Có thể sử dụng nhiều trường signature header entry cho một tên dữ liệu được gửi đi.

* + 1. Các ưu điểm của chữ kí số
* Khả năng xác định nguồn gốc

Các hệ thống mật mã hóa khóa công khai cho phép mật mã hóa văn bản với khóa bí mật mà chỉ có người chủ của khóa biết. Để sử dụng chữ ký số thì văn bản cần phải được mã hóa bằng hàm băm (văn bản được "băm" ra thành chuỗi, thường có độ dài cố định và ngắn hơn văn bản) sau đó dùng khóa bí mật của người chủ khóa để mã hóa, khi đó ta được chữ ký số. Khi cần kiểm tra, bên nhận giải mã (với khóa công khai) để lấy lại chuỗi gốc (được sinh ra qua hàm băm ban đầu) và kiểm tra với hàm băm của văn bản nhận được. Nếu 2 giá trị (chuỗi) này khớp nhau thì bên nhận có thể tin tưởng rằng văn bản xuất phát từ người sở hữu khóa bí mật. Tất nhiên là chúng ta không thể đảm bảo 100% là văn bản không bị giả mạo vì hệ thống vẫn có thể bị phá vỡ.

Vấn đề nhận thực đặc biệt quan trọng đối với các giao dịch tài chính. Chẳng hạn một chi nhánh ngân hàng gửi một gói tin về trung tâm dưới dạng (a,b), trong đó a là số tài khoản và b là số tiền chuyển vào tài khoản đó. Một kẻ lừa đảo có thể gửi một số tiền nào đó để lấy nội dung gói tin và truyền lại gói tin thu được nhiều lần để thu lợi (tấn công truyền lại gói tin).

* Tính toàn vẹn

Cả hai bên tham gia vào quá trình thông tin đều có thể tin tưởng là văn bản không bị sửa đổi trong khi truyền vì nếu văn bản bị thay đổi thì hàm băm cũng sẽ thay đổi và lập tức bị phát hiện. Quá trình mã hóa sẽ ẩn nội dung của gói tin đối với bên thứ 3 nhưng không ngăn cản được việc thay đổi nội dung của nó. Một ví dụ cho trường hợp này là tấn công đồng hình (homomorphism attack): tiếp tục ví dụ như ở trên, một kẻ lừa đảo gửi 1.000.000 đồng vào tài khoản của a, chặn gói tin (a,b) mà chi nhánh gửi về trung tâm rồi gửi gói tin (a,b3) thay thế để lập tức trở thành triệu phú!Nhưng đó là vấn đề bảo mật của chi nhánh đối với trung tâm ngân hàng không hẳn liên quan đến tính toàn vẹn của thông tin gửi từ người gửi tới chi nhánh, bởi thông tin đã được băm và mã hóa để gửi đến đúng đích của nó tức chi nhánh, vấn đề còn lại vấn đề bảo mật của chi nhánh tới trung tâm của nó

* Tính không thể phủ nhận

Trong giao dịch, một bên có thể từ chối nhận một văn bản nào đó là do mình gửi. Để ngăn ngừa khả năng này, bên nhận có thể yêu cầu bên gửi phải gửi kèm chữ ký số với văn bản. Khi có tranh chấp, bên nhận sẽ dùng chữ ký này như một chứng cứ để bên thứ ba giải quyết. Tuy nhiên, khóa bí mật vẫn có thể bị lộ và tính không thể phủ nhận cũng không thể đạt được hoàn toàn.

* + 1. Thực hiện chữ kí số khóa công khai

Chữ ký số khóa công khai dựa trên nền tảng mật mã hóa khóa công khai. Để có thể trao đổi thông tin trong môi trường này, mỗi người sử dụng có một cặp khóa: một công khai và một bí mật. Khóa công khai được công bố rộng rãi còn khóa bí mật phải được giữ kín và không thể tìm được khóa bí mật nếu chỉ biết khóa công khai.

**Toàn bộ quá trình gồm 3 thuật toán:**

+ Thuật toán tạo khóa

+ Thuật toán tạo chữ ký số

+ Thuật toán kiểm tra chữ ký số

**Ví dụ:**

Bob muốn gửi thông tin cho Alice và muốn Alice biết thông tin đó thực sự do chính Bob gửi. Bob gửi cho Alice bản tin kèm với chữ ký số. Chữ ký này được tạo ra với khóa bí mật của Bob. Khi nhận được bản tin, Alice kiểm tra sự thống nhất giữa bản tin và chữ ký bằng thuật toán kiểm tra sử dụng khóa công cộng của Bob. Bản chất của thuật toán tạo chữ ký đảm bảo nếu chỉ cho trước bản tin, rất khó (gần như không thể) tạo ra được chữ ký của Bob nếu không biết khóa bí mật của Bob. Nếu phép thử cho kết quả đúng thì Alice có thể tin tưởng rằng bản tin thực sự do Bob gửi.

Thông thường, Bob không mật mã hóa toàn bộ bản tin với khóa bí mật mà chỉ thực hiện với giá trị băm của bản tin đó. Điều này khiến việc ký trở nên đơn giản hơn và chữ ký ngắn hơn. Tuy nhiên nó cũng làm nảy sinh vấn đề khi 2 bản tin khác nhau lại cho ra cùng một giá trị băm. Đây là điều có thể xảy ra mặc dù xác suất rất thấp.

* + 1. Một vài thuật toán chữ kí số
* Full Domain Hash, RSA-PSS ..., dựa trên RSA
* DSA
* ECDSA
* ElGamal signature scheme
* Undeniable signature
* SHA (thông thường là SHA-1) với RSA

1. ***Giới thiệu chứng chỉ bảo mật X509***
   1. *Tổng quan*

X.509 là một đề nghị của ITU (International Telecommunication Union) định nghĩa một framework về chứng thực (certificate). X.509 dựa trên X.500, mà bản thân X.500 còn chưa được định nghĩa hoàn hảo. Kết quả là chuẩn X.509 đang được diễn giải theo một số cách, tùy theo công ty cung cấp quyết định sử dụng như thế nào. X.509 lần đầu tiên được công bố vào năm 1988, và các phiên bản tiếp theo đã được đưa ra để giải quyết các vấn đề an toàn. X.509 hỗ trợ cả hai mã bí mật (mã đơn) và mã công khai. X.509 định nghĩa các nội dung về một chứng thực, bao gồm số phiên bản, số serial, ID chữ ký, tên công bố, thời điểm có hiệu lực, định nghĩa chủ đề, phần mở rộng và chữ ký trên các trường trên. Về cơ bản, một người có trách nhiệm chứng nhận sẽ đặt khóa công khai của một người nào đó có nhu cầu chứng thực vào thủ tục chứng thực và sau đó xác thực lại bằng khóa riêng. Điều nầy bắt buộc khóa và thủ tục chứng thực phải luôn đi kèm với nhau. Bất cứ ai cần dùng khóa công cộng của một đối tượng nào đó đều có thể mở thủ tục chứng thực bằng khóa công cộng của các đối tượng nầy do người có trách nhiệm chứng thực cung cấp (các khóa công cộng nầy được ký hoặc khóa bằng khóa riêng của người có trách nhiệm chứng thực). Vì vậy, người sử dụng phải tin rằng người có trách nhiệm chứng thực sẽ bảo đảm việc hợp lệ hóa người chủ của khóa công khai và thực sự khóa công khai ở đây chính là khóa công khai của người có trách nhiệm chứng thực. Đây chính là nơi chứa các PKI (public-key infrastructures). PKI là một kiến trúc phân cấp những đối tuợng có trách nhiệm xác minh các khóa công khai lẫn nhau.

Cấu trúc trong chứng chỉ X509:

* Certificate
  + Version
  + Serial Number
  + Algorithm ID
  + Issuer
  + Validity

+ Not Before

+ Not After

* + Subject
  + Subject Public Key Info

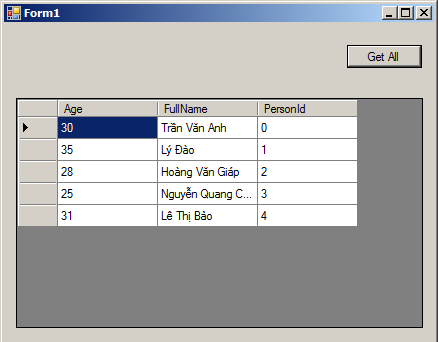
+ Public Key Algorithm

+ Subject Public Key

* + Issuer Unique Identifier (optional)
  + Subject Unique Identifier (optional)
  + Extensions (optional)
* Certificate Signature Algorithm
* Certificate Signature
  1. *Triển khai thực tế*

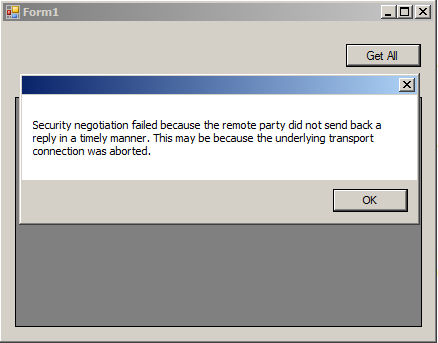
Kết quả của quá trình triển khai kiểm thử chứng chỉ bảo mật X509 trên máy nội bộ mô phỏng:

* Quá trình xác thực thành công



Sau khi xác thực thành công thì màn hình kết quả sẽ hiển thị

* Quá trình xác thực thất bại: thông báo sẽ hiện ra



* + - * File XML lưu lại thông tin về quá trình thực hiện xác thực thông tin:



* + - * Thông tin xác thực trên Client:



* + - * Thông tin xác thực trên Server:



1. **Lập trình kiểm thử**
2. ***Đặt vấn đề***

Hiện nay, có rất nhiều ứng dụng hoạt động trên môi trường web (Web-based Application). Các giao dịch được thực hiện thường xuyên giữa các bên thông qua các dịch vụ mà giao thức HTTP cung cấp (chủ yếu). Bài tiểu luận này tập trung nghiên cứu Web Service được thực thi trên giao thức HTTP, tổ chức theo mô hình SOAP (Simple Object Access Protocol).

Các bên khi giao dịch sẽ sử dụng các thông điệp (message) để trao đổi thông tin. Chính vì thế, việc đảm bảo an toàn và bảo mật cho các gói tin này là nhiệm vụ hết sức quan trọng. Bởi lẽ, hiện nay có rất nhiều hình thức tấn công của các hacker nhằm lấy trộm thông tin trong các thông điệp cho các mục đích xấu như: message alteration, man-in-the-middle, replay attacks... Các nhà phát triển hiện nay đã đưa ra những giải pháp phòng chống rất hiệu quả như: các cơ chế xác thực, phân quyền, mã hóa dữ liệu, xử dụng chữ kí số để xác định danh tính...

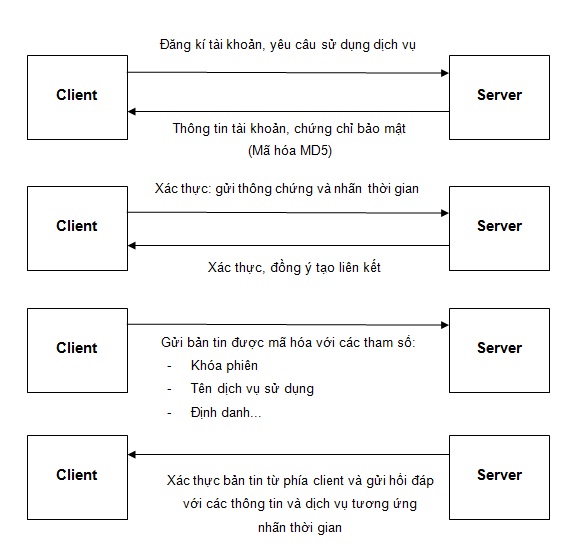
Ý tưởng của nhóm mong muốn lập trình kiểm thử một trong những giải pháp trên để củng cố vững chắc kiến thức về bảo mật trong Web Service. Nhóm đã thống nhất và lựa chọn kiểm thử trên cơ chế xác thực thông tin người dùng (Authentication)

1. ***Mục tiêu***
   * + - Hiểu rõ các thành phần, tổ chức của Web Service sử dụng trong giao thức HTTP và các vấn đề liên quan
       - Nắm được cơ chế bảo mật trong Web service (Security in Web Service)
       - Cập nhật thông tin về các hình thức tấn công hướng vào thông điệp người dùng
       - Hiểu rõ về các giải pháp chống tấn công hiệu quả
       - Hiểu rõ về các thuật toán được sử dụng trong các giải pháp đó
       - Đưa ra được đánh giá về các giải pháp cụ thể đó
2. ***Thực nghiệm***
   1. *Phân tích yêu cầu*
      * + Cung cấp danh sách và thông tin chi tiết về đội ngũ nhân viên của công ty.
        + Đối với mỗi đối tượng nhân viên sẽ có những quyền hạn cụ thể đối với hệ thống.

|  |  |  |  |
| --- | --- | --- | --- |
| Quyền hạn | Ban giám đốc | Quản lý nhân viên | Nhân viên |
| Xem danh sách nhân viên | OK | OK | OK |
| Tìm kiếm | OK | OK | OK |
| Xem chi tiết | OK | OK | OK |
| Sửa đổi thông tin | OK | OK | NOT |
| Xóa nhân viên | OK | NOT | NOT |
| Chỉnh sửa quyền hạn | OK | NOT | NOT |

Bảng phân quyền người dùng của hệ thống

* 1. *Môi trường*
* Máy chủ windows server 2008
* Web server: internet information services 7.0
* Nền tảng ứng dụng: .NET framework 4.0
  1. *Chức năng*
* Xem danh sách nhân viên
* Tìm kiếm nhân viên + Xem thông tin chi tiết của một nhân viên cụ thể
* Sửa đổi thông tin của một nhân viên
* Xóa một nhân viên khỏi danh sách
* Chỉnh sửa quyền hạn của một nhân viên
  1. *Các yêu cầu về bảo mật*
* Đảm bảo chính xác quyền hạn với các nhân viên
* Chống các hình thức tấn công giả mạo, chặn giữ thông tin, sửa đổi thông tin
* Chống tấn công thụ động
  1. *Các hình thức áp dụng*
* Xác thực đối với người dùng.
* Bảo mật kênh truyền bằng Secure Socket Layer (SSL) và Mã hóa bản tin bằng chứng chỉ X509.
  1. *Kịch bản áp dụng bảo mật*
* Trường hợp: client muốn sử dụng một dịch vụ nào đó của server
* Các bước:
  + Bước 0: client đăng ký tài khoản và các yêu cầu sử dụng dịch vụ với Server, Server đồng ý và gửi về cho client thông tin tài khoản và chứng chỉ bảo mật. Mật khẩu của client được mã hóa sử dụng bảng băm MD5. Với mỗi một chuỗi mật khẩu có độ dài bất kỳ, bảng băm MD5 sẽ cho ra một chuỗi có độ dài không đổi 128 bit và duy nhất với các chuỗi đầu vào khác nhau. Người quản trị cũng như bên tấn công không thể giải mã được chuỗi mật khẩu đã mã hóa.
  + Bước 1: Khi client muốn sử dụng một dịch vụ của server, client và server sẽ cùng nhau xác thực thông qua chứng chỉ ở phía server và chứng chỉ được cấp phát cho client. Các chứng chỉ này đều có gốc là một chứng chỉ được tin cậy đặt trên máy server. Các bước xác thực bằng chứng chỉ:
    - Client gửi bản tin yêu cầu kết nối cùng bản tin mã hóa bao gồm thông chứng của mình kèm theo nhãn thời gian cho server.
    - Server nhận và giải mã bản tin, sau khi xác nhận chứng chỉ của client cùng nhãn thời gian là hợp lệ, server đồng ý tạo liên kết với client.
  + Bước 2: Client gửi một bản tin được mã hóa mô tả dịch vụ muốn sử dụng, các tham số cùng thông tin bảo mật gồm có:
    - Khóa phiên:
      * Nhãn thời gian.
      * Khóa công khai của client.
    - Tên dịch vụ sử dụng.
    - Các tham số dịch vụ.
    - Định danh (Username – trên truy cập) client đã đăng ký trước đó.
    - Mật khẩu của client.
  + Bước 3: Server nhận và giải mã bản tin, xác thực người dùng hợp lệ và có quyền sử dụng dịch vụ. Server thực hiện dịch vụ dựa trên các danh sách tham số mà người dùng đã gửi tính toán và gửi trả kết quả cho client dưới dạng bản tin đã được mã hóa. Kết quả bao gồm
    - Khóa phiên
      * Nhãn thời gian.
      * Khóa công khai của server.
    - Nhãn thời gian trong bản tin mà client đã gửi trước đó.
    - Tên dịch vụ sử dụng.
    - Các tham số dịch vụ.
    - Kết quả thực hiện dịch vụ.
  + Bước 4: Client nhận bản tin hồi đáp, xác thực người gửi là server cùng dựa vào các thông tin như nhãn thời gian của bản tin yêu cầu, tên dịch vụ, các tham số để đảm bảo rằng kết quả trả về là chính xác. Client gửi lại cho server một bản tin acknowledgement xác nhận lại kết quả là chính xác và tin cậy. Kết thúc quá trình thực hiện một dịch vụ.



Các bước trong quá trình xác thực giữa client và server

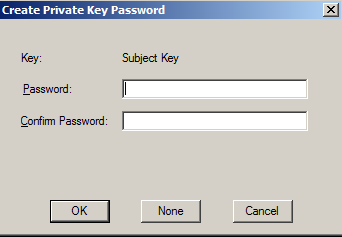
* 1. *Các bước triển khai*
* Xây dựng và triển khai ứng dụng cung cấp web services với các chức năng tương ứng, ứng dụng có tên SecurceContact, chạy trên IIS 7.0
  + Xem toàn bộ danh sách nhân viên.
  + Tìm kiếm nhân viên theo ID và xem chi tiết của một nhân viên cụ thể.
  + Sửa đổi thông tin của nhân viên.
  + Xóa nhân viên khỏi danh sách.
  + Chính sửa quyền hạn của nhân viên.

Bên cạnh các chức năng, ứng dụng phải đảm bảo các quy trình bảo mật đã nêu ở trên.

* Cấu hình ứng dụng sử dụng SSL
  + Tạo chứng chỉ bảo mật theo X509 gốc RootCA
  + Vào Command Prompt. Gõ dòng lệnh

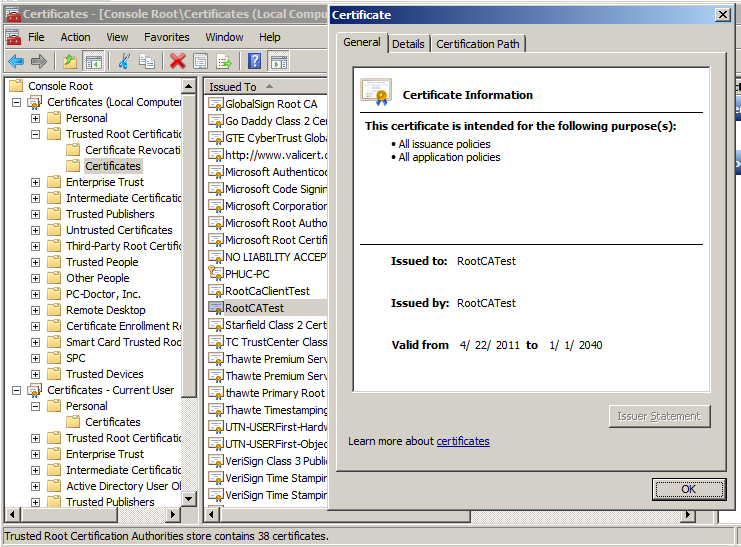
makecert ‐n "CN=RootCATest" ‐r ‐sv RootCATest.pvk RootCATest.cer.

* + Gõ mật khẩu, đây có thể coi là khóa chung chia sẻ giữa client và server.



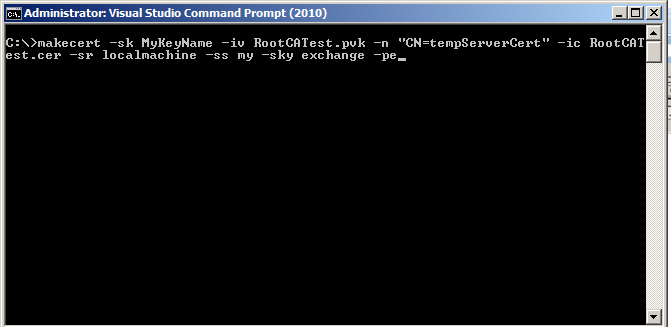
Sau khi thực hiện, ta có 2 file "RootCATest.pvk" và "RootCATest.cer".

* + Tạo  Certificate Revocation List từ các file đã làm ở bước trên bằng cách gõ dòng lệnh trong Command Prompt
    - * + makecert ‐crl ‐n "CN=RootCATest" ‐r ‐sv RootCATest.pvk RootCATest.crl
  + Tiến hành cài đặt lên server
    - * + Mở Microsoft Management Console bằng cách vào Run, gõ mmc.
        + Cài đặt chứng chỉ vào Local Machine 🡪 Trusted Root Certification Authorities 🡪Certificates

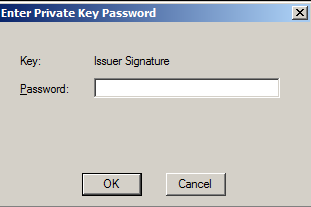


* + Tạo chứng chỉ phía server
    - Dựa trên những file chứng chỉ đã có, tạo chứng chỉ dựa trên RootCATest, gõ đoạn lệnh sau trong Command Prompt makecert ‐sk MyKeyName ‐iv RootCATest.pvk ‐n "CN=tempServerCert" ‐ic

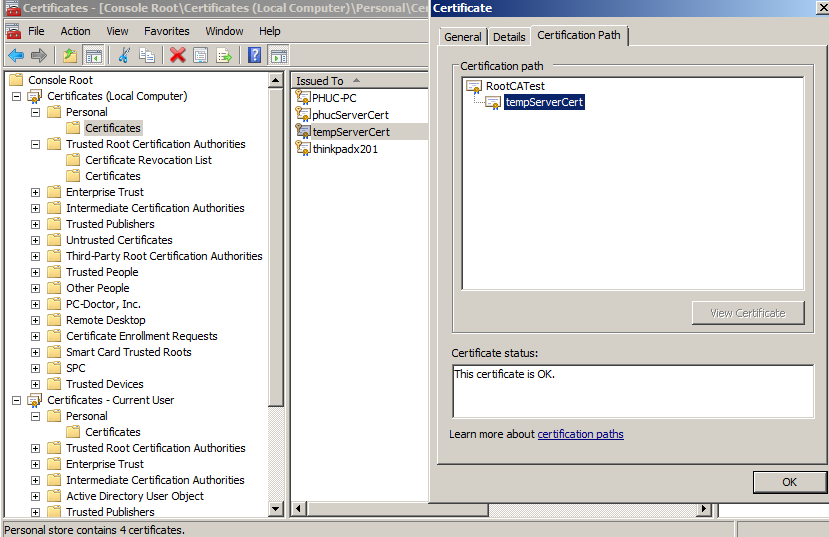
RootCATest.cer ‐sr localmachine ‐ss my ‐sky exchange ‐pe



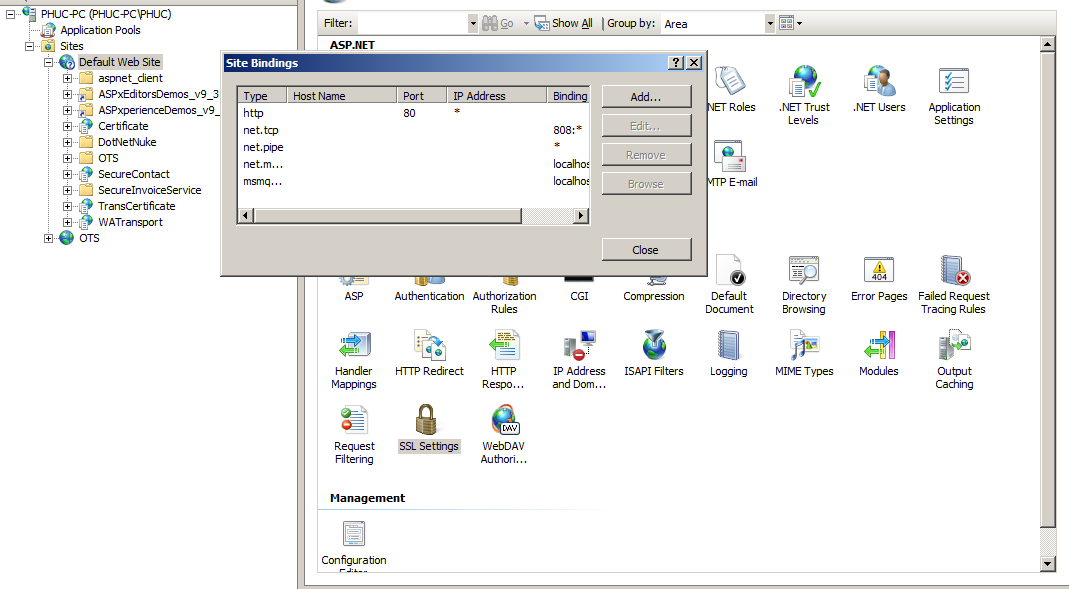
* + - Điền mật khẩu, mật khẩu này phải giống với mật khẩu của RootCATest



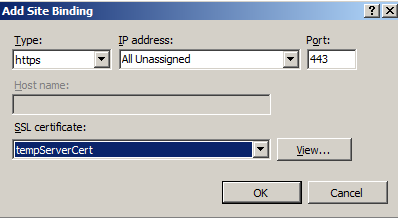
* + - Sau khi tạo chứng chỉ thành công, chứng chỉ sẽ được cài đặt vào các chứng chỉ của LocalMachine 🡪 Personal 🡪 Certificates



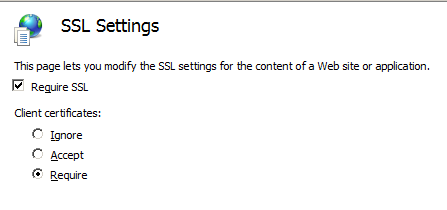
* + - Cấu hình ứng dụng sử dụng SSL
      * Sau khi đã cài đặt chứng chỉ cho server, iis cũng sử dụng chứng chỉ đó để thực hiện bảo mật. Cấu hình chứng chỉ cho website, chọn mục Bindings trong website.



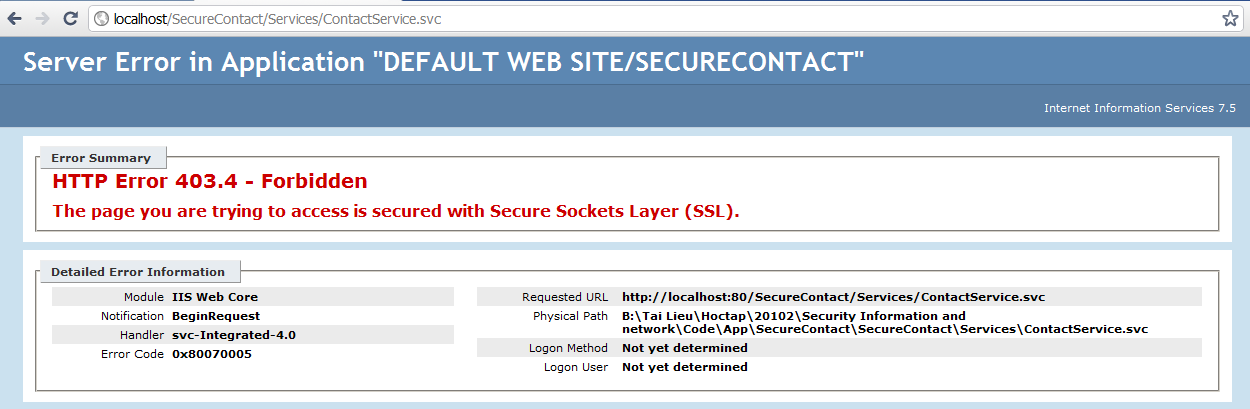
* + - * Chọn Add, cấu hình https và chọn chứng chỉ vừa cài đặt.



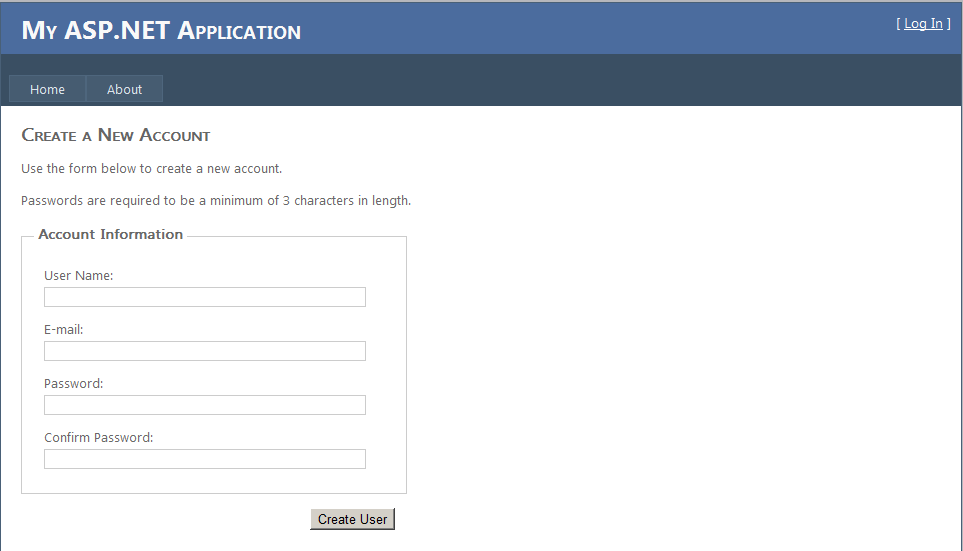
* + - * Cấu hình yêu cầu website buộc phải truy cập qua SSL, vào mục SSL Setting, chọn Require SSL



Sau khi cấu hình xong, người dùng không thể truy cập địa chỉ của trang thông qua phương thức bình thường, trang báo lỗi như sau



* Xây dựng ứng dụng phía client.
  + Để ứng dụng có thể sử dụng dịch vụ của server cung cấp, trước tiên người dùng phải đăng ký với ứng dụng.



* + Ứng dụng sẽ cấp cho client một chứng chỉ dựa trên chứng chỉ RootCATest. VD: tempClientCert. Phía client phải cài đặt chứng chỉ thông qua Microsoft Management Console vào: CurrentUser (tài khoản người dùng hiện tại trên Windows) 🡪 Personal 🡪 Certificates.

1. ***Đánh giá và kết luận***
   1. *Đánh giá*

* Kết quả của quá trình thực nghiệm đã thành công và đáp ứng được các yêu cầu đã đặt ra
* Việc xác thực với mỗi đối tượng người dùng được đảm bảo trong quá trình tương tác của người dùng với hệ thống
* Kết quả thông báo hiển thị trên màn hình trực quan,dễ hiểu giúp cho đội ngũ kĩ thuật và cả người dùng dễ dàng nhận thấy lỗi phát sinh.
* Toàn bộ quy trình thực hiện (flow) được đảm bảo ràng buộc, khoa học, không gây nhầm lẫn.
  1. *Kết luận*
* Việc áp dụng triển khai phương thức xác thực (Authentication) và mã hóa bảo mật (Encryption) trên web hoàn toàn khả thi.
* Hiệu quả của việc sử dụng các phương thức đảm bảo an ninh này rất cao.

**KẾT LUẬN**

Web Service đang trở thành một phương thức giao tiếp trên mạng phổ biến và hiệu quả. Cách thức tổ chức của Web Service rất linh động, phù hợp với nhiều môi trường hay các đối tượng sử dụng. Các cơ chế tham gia trong quá trình chuyển vận thông tin của Web Service cũng mạnh mẽ để đảm bảo quá trình trao đổi dữ liệu diễn ra thành công, hiệu quả.

Mặc dù vậy, Web Service không thể tránh khỏi những lỗ hổng bảo mật để tạo cơ hội cho kẻ xấu xâm nhập, tấn công vào hệ thống. Hiện nay, việc tìm hiểu về vấn đề bảo mật trong Web Service không còn mới, rất nhiều phương pháp tấn công đã được phát hiện và ngăn chặn kịp thời. Tuy nhiên, vấn đề bảo mật an toàn thông tin trên mạng vẫn luôn diễn ra hàng ngày hàng giờ, rất cần những giải pháp an ninh hiệu quả hơn nữa.

Sau khi nghiên cứu tìm hiểu đề tài về bảo mật trong Web Service, nhóm sinh viên chúng em đã tự trang bị thêm nhiều kiến thức về vấn đề an toàn bảo mật thông tin. Có nhiều vấn đề khó khăn chúng em chưa thể giải quyết được trong phạm vi bài tiểu luận. Chúng em mong tiếp tục nhận được sự giúp đỡ, hướng dẫn của PGS.TS. Nguyễn Linh Giang để cải thiện thêm về những vấn đề này.

**TÀI LIỆU THAM KHẢO**

[1] **Web Services Architecture W3C Working Group Note 11 February 2004**

*Editors:*

David Booth, W3C Fellow / Hewlett-Packard

Hugo Haas, W3C

Francis McCabe, Fujitsu Labs of America

Eric Newcomer (until October 2003), Iona

Michael Champion (until March 2003), Software AG

Chris Ferris (until March 2003), IBM

David Orchard (until March 2003), BEA Systems

[2] **Web Services Security Kerberos Binding Working Draft 01, 18 September 2002**

*Editors:*

Phillip Hallam-Baker, VeriSign

Chris Kaler, Microsoft

Ronald Monzillo, Sun

Anthony Nadalin, IBM

[3] Web Service – Wikipedia

<http://en.wikipedia.org/wiki/Web_service>

[4] Web Service Specifications – MSDN Center

<http://msdn.microsoft.com/en-us/library/ms951274.aspx>

[5] Digital Signing Concept – W3C

<http://www.w3.org/TR/2001/NOTE-SOAP-dsig-20010206/>