THỰC THI BẢO MẬT

Mục tiêu bài học:

Cuối chương này bạn có thể

- Mô tả về công cụ JAR
- Tạo và xem một file JAR, và liệt kê và trích rút nội dung của file.
- Sử dụng chữ ký điện tử (Digital Signatures) để nhận dạng Applets
- Tạo bộ công cụ khóa bảo mật (Security key)
- Làm việc với chứng chỉ số (Digital Certificate)
- Tìm hiểu về gói Java.security

10.1 Giới thiệu:

Trong phần này, chúng ta sẽ tìm hiểu chi tiết về bảo mật Java applet. Chúng ta cũng thảo luận về mô hình bảo mật JDK 1.2 đáp ứng nhu cầu người dùng và nhà phát triển.

Java là một ngôn ngữ lập trình đầu tiên gởi các chương trình không tương tác như các file văn bản, file ảnh và các thông tin tĩnh thông qua World Wide Web. Các chương trình này, không giống như chương trình CGI, được chạy trên hệ thống của người dùng, hơn là chạy trên máy chủ Web (Web server). Bảo mật Java Applet là sự quan tâm chính giữa người dùng và nhà phát triển applet. Thiết tính bảo mật trong applet có thể dẫn tới sửa đổi hoặc phơi bày các dữ liệu nhạy cảm. Mô hình bảo mật của Java 2, hoặc JDK 1.2 rất hữu ích cho người dùng, cũng như cho nhà phát triển. Nó giúp người dùng duy trì mức độ bảo mật cao. Trong chương nay, chúng ta sẽ học mô hình bảo mật JDK 1.2.

10.2 Công cụ JAR:

Một file JAR là một file lưu trữ được nén do công cụ lưu trữ Java tạo ra. File này tương tự như chương trình PKZIP. Nó chứa nhiều file trong một file lưu trữ. Điều này cho phép tải trong trình duyệt hiệu quả. Dùng một jar với một applet cải tiện đáng kể khả năng thực hiện của trình duyệt. Vì tấc cả các tất cả các file được biên dịch trong một file đơn, trình duyệt chỉ cần thiết lập kết nối HTTP với web server. Nén file giảm 50% thời gian tải file.

Để khởi động công cụ JAR, dùng câu lệnh sau tại dấu nhắc lệnh:

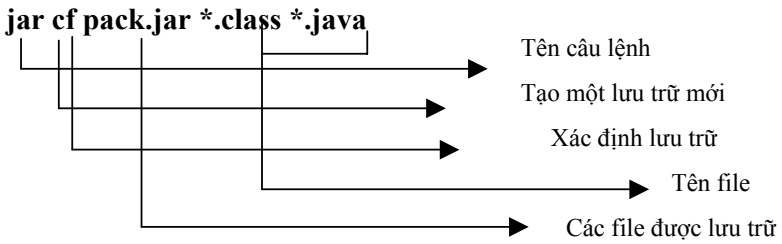
jar [options][manifest] jar-file input-file(s)

Tuỳ chọn	Mô tả
c	Tạo ra một lưu trữ mới
t	Ghi vào bảng nội dung cho lưu trữ
x	Trích dẫn file có tên từ lưu trữ
v	Tạo nguồn xuất đa dòng (verbose output) trên một lỗi chuẩn
f	Xác định tên file lưu trữ
m	Bao hàm thông tin chứng thực từ các file chứng thực xác định
o	Lưu trữ chỉ 'use no zip' nén
M	Không tạo các file chứng thực cho các mục (entries).

Bảng 0.1. công cụ jar

Một file chứng thực chứa thông về các file lưu trữ. File này là một tuỳ chọn. Thậm chí file không xác định thì JAR cũng tự động tạo ra. File jar được dùng như các lưu trữ. File này phải có phần mở rộng là '.jar' được xác định tại dòng lệnh. File đầu vào (input-file) là danh sách phân cách các file được đặt trong lưu trữ. Netscape Navigator và Internet Explorer hỗ trợ file JAR.

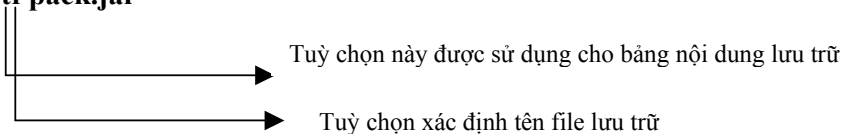
Câu lệnh sau lưu trữ tất cả các file class và file java bao gồm trong một thư mục xác định vào một file jar gọi là 'pack'



Hình 10.1 lệnh jar

Dùng lệnh sau tại dấu nhắc liệt kê các file trong file 'pack.jar'

jar tf pack.jar



Hình 10.2 Liệt kê các file trong file pack.jar

Để gộp file lưu trữ 'pack.jar vào trong một applet, mở trang HTML, và thêm thuộc tính ARCHIVE='pack.jar' vào thẻ applet, như sau:

<applet code="exr7.class" ARRCHIVE="pack.jar" height=125 width=350></applet>

Thuộc tính sẽ chỉ cho trình duyệt nạp lưu trữ 'pack.jar' để tìm file 'exr7.class'

Câu lệnh sau trích rút các file được nén trong file pack.jar:

jar xvf pack.jar

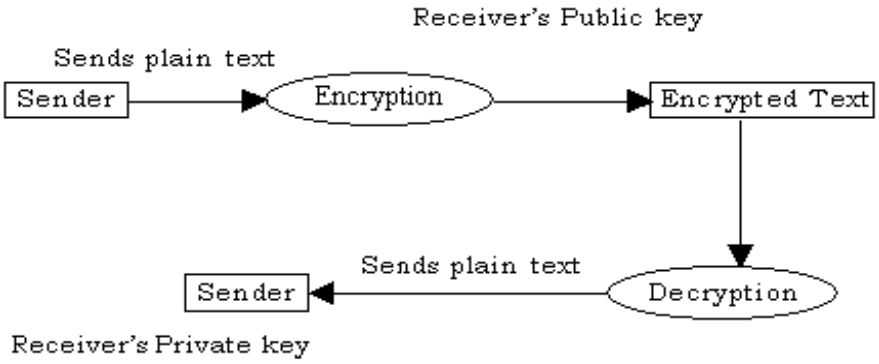
Mục chọn 'x' cho phép bạn trích rút nội dung của file.

10.3 Chữ ký điện tử (Digital Signature) cho định danh các applet:

Trong java, bảo mật applet trên web là phần rất quan trọng. Hacker có thể viết các applet nguy hiểm xuyên thủng hàng rào bảo mật. Vì thế, applet hạn chế sự can thiệt của các ngôn ngữ. Applet không hỗ trợ một số nét đặt trưng sau:

- Đọc và ghi file từ hệ thống nơi applet đang chạy.
- Lấy thông tin về một file từ hệ thống
- Xoá một file từ hệ thống.

Java 2 có thể thực hiện tất cả các đặc điểm trên, với các applet cung cấp từ một nhà cung cấp applet tin cậy, và được ký danh số (digitally signed).
Hình sau minh họa quá trình mã hoá khoá



Hình 10.3. Mã hoá dựa trên các khoá

Trong hình trên, khoá công cộng (public keys) được dùng mã hoá và giải mã. Cùng ý tưởng được sử dụng cho chữ ký số, thêm các tính năng bổ sung.
Một chữ ký số là một file mã hoá cung cấp chương trình nhận dạng chính xác nguồn gốc của file. Khóa bí mật tính giá trị từ file applet. Người giữ khoá bí mật kiểm tra nội dung của đối tượng.
Trong định danh số, một khóa riêng (private key) được sử dụng để mã hóa, và khoá công cộng, được dùng giải mã. Trong khi ký danh (sign) một đối tượng, người ký danh dùng thuật toán tóm lược thông báo như MD5 để tính bảng tóm lược của đối tượng. Bảng tóm lược được dùng như là dấu tay cho đối tượng. bảng tóm lược lần lượt được mã hoá dùng khóa riêng, đưa ra chử ký điện tử của đối tượng. Khoá công cộng của bộ ký duyệt dùng để mã hoá chữ ký và kiểm tra chúng. Kết quả của sự giải mã, giá trị tóm lược được đưa ra. Giá trị tóm lược của đối tượng được tính và so sánh với giá trị tóm lược được giải mã. Nếu giá trị tóm lược (digest) của đối tượng và giá trị tóm lược được mã hoá khớp với nhau, chữ ký được được xác nhận. Tài liệu mô tả chữ ký được gọi là "Chứng thực" (Certificate)
Thiết lập sự uỷ thác (trust), nhận dạng applet được chứng nhận. Chứng nhận các thực thể các sử dụng khóa công cộng đặt biệt. Quyền chứng thực (a certificate authority) được dùng thực hiện chứng nhân. Nhận được được chứng thực từ một CA (Certificate Authority), applet phải đệ trình tài liệu chứng thực sự nhận dạng của nó.
Hiện giờ các công ty đưa ra các dịch vụ xác nhận chứng thực sau:

- VeriSign
- Chứng thực Thawte

Bạn có thể thiết lập các mức bảo mật khác nhau. Một applet có thể đưa ra sự uỷ thác hoàn toàn, hoặc không uỷ thác, với sự giúp đỡ của tập các lớp gọi là "Quyền" (Permissions). Nhưng nhìn chung, mỗi applet được giới hạn một cách đầy đủ, trừ khi nhà phát triển ký danh applet. Điều này thiết lập cho nhà phát triển đáng tin cậy.

10.4 Khoá bảo mật Java (Java Security key).

Chúng ta cần tạo 3 công cụ, tên là, 'jar', 'jarsigner', và 'keytool', trước khi dùng các applet ký danh. Chúng ta cần tạo cặp khóa công cộng/riêng, và làm cho nó trở nên sẵn sàng với công cụ jarsigner.

Bây giờ, chúng ta sẽ tạo các công dụng của keystore.

- Keystore (Lưu trữ khoá)

  Keystore là một cơ sở dữ liệu khoá, chứa các chứng thực số dùng để nhận dạng các giá trị khoá công cộng.

- Keytool (Công cự khoá)

Keytool là công cụ khoá bảo mật của java, tạo và quản lý khóa công cộng, khoá riêng, và các chứng thực bảo mật. Nó cũng có thể thực hiện:

- Quản lý cặp khoá công cộng/riêng
- Lưu trữ các khoá công cộng
- Dùng các chứng thực để xác thực chứng thực khác.
- Xác thực (Authenticate) dữ liệu nguồn.

Tất cả thông tin mà keytool quản lý được lưu trữ trong cơ sở dữ liệu gọi là keystore. Sun có một keystore mật định dùng một định dạng file mới gọi là JKS (java key store Lưu trữ khoá java). Để kiểm nếu hệ thống bạn có một keystore dưới định dạng này, thực hiện câu lệnh sau tại dấu nhắc lệnh:

Keytool –list

Thông báo lỗi sau xuất hiện nếu bạn không có gì trong keystore của bạn.

Keytool error: keystore file does not exist: c:\windows\.keystore

JDK tìm keystore chính trong thư mục C:\windows\. Đây là một vị trí chung cho các file hệ thống quan trọng trên windows 95, 98 và NT systems.

Tuỳ chọn keystore cũng có thể được sử dụng trong lệnh keytool, như sau:

keytool –list keystore c:\java\try

Câu lệnh này chỉ cho JDK tìm keystore trong file được gọi là 'try' trong thư mục 'C:\java\try'. Nếu không tìm thấy, sẽ hiển thị thông báo lỗi như trên.

Mục '-genkey' có thể được sử dụng cùng với câu lệnh keytool để tạo cặp khoá công cộng/riêng. Bạn cũng có thể dùng một số các tuỳ chọn khác. Dạng đơn giản nhất như sau:

keytool –genkey –alias "I"

Bí danh (alias) có thể được dùng lưu trữ, thay thế hoặc xoá cặp khoá. Các bí danh keytool không phân biệt chữ hoa. Trong lệnh trên, chúng ta không sử dụng tuỳ chọn keystore. Nếu cùng câu lệnh sử dụng tuỳ chọn keystore, sẽ được viết lại như sau:

keytool –genkey –alias "I" –keystore "store"

Trong lệnh trên, cặp khoá sẽ được lưu trữ trong keystore 'store', và không lưu trong keystore mật định của hệ thống.

Sau khi nhập lệnh trên vào, và nhấn phím enter, keytool nhắc bạn nhập vào mật khẩu (password) cho keystore, như sau:

Enter keystore password

Nhập vào 'password' như yêu cầu.
Tiếp theo, keytool nhắc bạn nhập vào các thông tin bổ sung như:

What is your first and last name? (Tên và họ)
[unknown]
what is the name of your organization unit?

[unknown]: software Development.
What is the name of your organization? (Tên của tổ chức)
[Unknown]: ABC Consultants (tư vấn ABC)
What is the name of your city or Locality? (tên thành phố hoặc địa phương của bạn)
[Unknown]: California
What is the name of your State or Province? (tên bang hoặc tỉnh của bạn)
[Unknown]:United States of America
What is the two-letter country code for this unit?(Mã quốc gia với 2 ký tự)
[Unknown]: US

Khi bạn đã nhập vào các thông tin, keytool hiển thị thông tin sau:

Is <CN=Bob Fernandes, OU=Software Development, O=ABC Consultants, L=California, ST=United States of America, C=US>correct?
[no]:

Cuối cùng, keystool nhắc bạn nhập vào mật khẩu cho khoá riêng của bạn, như:

Enter key password for <I>
(RETURN if same as keystore password)

Thông tin trên được sử dụng để kết hợp sự phân biệt tên (name) X500 với bí danh (alias). Thông tin trên cũng có thể được đưa vào trực tiếp từ mục chọn '-dname'

Mật khẩu sau cùng phần biệt với mật khẩu keystore. Nó được dùng truy cập khoá riêng của cặp khoá công cộng. Mật khẩu có thể trực tiếp chỉ rõ bằng cách sử dụng tuỳ chọn '-keypass'. Nếu mật khẩu không chỉ rõ, mật khẩu keystore được sẽ được dùng. Tuỳ chọn '-keypasswd' dùng thay đổi mật khẩu. Tuỳ chọn '-keyalg' chỉ rõ thuật toán tạo cặp khoá.

Khi bạn tạo một khoá và bổ sung nó vào trong keystore, bạn có thể dùng tuỳ chọn '-list' của keytool để xem khoá có trong keystore hay không.

Để xoá cặp khoá từ cơ sớ liệu, dùng lệnh sau:

keytool –delete –alias aliasName

'aliasName' chỉ tên của khoá được xoá.

Bây giờ, chúng ta tạo cặp khoá riêng/công cộng cho file JAR, chúng ta hãy ký danh nó. Lệnh jarsigner dùng để ký danh một file JAR. Nhập lệnh sau vào dấu nhắc DOS:

jarsigner –keystore keyStore –storepass storePassword –keypass keyPassword

Bảng sau cung cấp danh sách của JARFileNames và bí danh:

Tuỳ chọn	Mô tả
keyStore	Tên keystore sử dụng
storePassword	Mật khẩu keystore
keyPassword	Mật khẩu khoá riêng
JARFileName	Tên của file JAR được ký danh
Alias	Bí danh của bộ ký danh

Bảng 10.2 JARFileNames và bí danh

Để ký danh file JAR 'pack.jar', với keystore 'store', và mật khẩu để lưu trữ và các khoá riêng là 'password', dùng lệnh sau:

jarsigner –keystore store –storepass password –keypass password pack.jar pk
'pk' nghĩa là tên bí danh.

Nếu tuỳ chọn '-keystore' không chỉ rõ, thì keystore mật định được dùng.

Để chỉ rõ chữ ký của file JAR được định danh, dùng tuỳ chọn '-verify'.

jarsigner –verify pack.jar

'pack.jar' chỉ tên file JAR. Nếu chữ ký không hợp lệ, thì ngoại lệ sau được ném ra (thrown).

Jarsigner:java.util.zip.ZipException:invalid entry size (expected 900 but got 876 bytes)

Ngược lại, xuất hiện thông báo "jar verified" (jar được xác minh)

Quá trình xác thực kiểm tra theo các bước sau:

- Có file '.DSA' chứa chữ ký hợp lệ cho file chữ ký .SF không.
- Có các mục trong file chữ ký là các tóm lược hợp lệ cho mỗi mục tương ứng file kê khai (manifest file)

10.5 Chữ ký điện tử (digital Certificates)

Cho đến bây giờ, chúng ta đã học cách tạo và ký danh một file JAR. Bây giờ, chúng ta sẽ học cách xuất các chữ ký điện tử(digital certificates), sẽ sử dụng để xác thực chữ ký của các file JAR. Chúng ta cũng sẽ học các nhập chữ ký điển tử từ các file các.

Chữ ký điện tử là một file, một đối tượng, hoặc một thông báo được ký danh bởi quyền chứng thực (certificate authority). The CA (Certificate authority) cấp chứng nhận giá trị các khoá công cộng. Chứng nhận X.509 của tổ chức International Standards Organization là một dạng chứng nhận số phổ biến. Keytool hổ trợ những chứng nhận này.

Keytool ở bước đầu tiên cần nhận được một chứng nhận (certificate). Chúng ta dùng chứng nhận đó tạo cặp khoá 'công cộng/riêng' (private/public). Keytool nhập vào các chứng nhận đã được tạo và được ký danh. Keytool tự động gắn (bundle) khoá công cộng mới với một chứng nhận mới. Cùng thực thể đã tạo khoá công cộng ký danh chứng nhận này. Đó được gọi là **'self-signed certificates'** (Chứng nhận tự ký danh).Các chứng nhận này không phải là chứng nhận đáng tin cậy cho định danh. Tuy nhiên, chúng cần để tạo các yêu cầu ký danh chứng nhận (certificate-signing request).

Keytool và tuỳ chọn được sử dụng để tạo các chứng nhận trên. Câu lệnh sau giúp tạo các chứng nhận trên:

keytool –keystore store –alias mykey –certreq –file mykey.txt

Cặp khoá được tạo là 'mykey'. Tuỳ chọn '-file' chỉ tên file, mà yêu cầu ký danh chứng nhận dùng để lưu.

Dùng lệnh '-export' xuất các chứng nhận này như sau:
keytool –export –keystore store –alias pk –file mykey

Câu lệnh trên hiển thị dấu nhắc sau:

Enter keystore password

Chứng nhận đã lưu trữ trong <mykey>

Để nhập các chứng nhận khác vào keystore của bạn, nhập câu lệnh sau:

keytool import –keytool keystore –alias alias –file filename

Tên được chỉ như là tên file chứa chứng nhận được nhập vào (imported certificate).

Câu lệnh sau chỉ tên bí danh là 'alice' để nhập chứng nhận trong file 'mykey' vào keystore 'MyStore':

keytool –import –keystore MyStore –alias alice –file mykey

Câu lệnh trên hiển thị dấu nhắc sau:

Enter keystore password (Nhập vào mật khẩu keystore)

Kết quả xuất ra hiển thị hai tuỳ chọn –Owner và Issuer. Nó hiển thị tên công ty, nghề nghiệp, tổ chức, địa điểm, bang và tiền tệ. Nó cũng hiển thị số serial và thời gian có giá trị. Cuối cùng, nó hỏi có là chứng nhận uỷ thác không. Chứng nhận được chấp thuận cho sự uỷ thác của riêng bạn.

Dùng lệnh '-list' liệt kê nội dung của keystore như sau:

keystool –list –keystore Store

Câu lệnh trên yêu yều password keystore

Dùng tuỳ chọn '-alias' liệt kê một mục. Dùng lệnh -delete để xoá bí danh trong keystore, như sau:

keytool –delete –keystore Store –alias alias

Dùng lệnh '-printcert' in chứng nhận được lưu trữ trong file, theo cách sau:

keytool –printcert –file myfile

Dùng lệnh '-help' nhận về danh sách tất cả các lệnh keytool hổ trợ:

keytool -help

10.6 Các gói bảo mật java (JAVA Security packages)

Các gói bảo mật Java bao gồm:

- java.security

Đây là gói API nhân bảo mật (the core security API package). Chứa các lớp và giao diện (interface) hỗ trợ mã hoá (encryption), tính bảng tóm lược tài liệu và chữ ký điện tử.

- java.security.acl

Chứa các giao diện dùng cài đặt các chính sách điều kiển truy cập

- java.security.cert

Cung cấp sự hổ trợ cho chứng nhận X.509

- java.security.interfaces

Định nghĩa các giao diện truy cập thuật toán chữ ký điện tử (the digital signature algorithm)

- java.security.spec

Cung cấp các lớp độc lập và phục thuộc vào thuật toán cho các khoá.

Tóm tắt:

- Nếu khả năng bảo mật trong applet không đảm bảo, các dữ liệu nhạy cảm có thể được sữa đổi hoặc phơi bày.
- Mục đích chính của JAR là kết nối các file mà applet sử dụng trong một file nén đơn. Điều này cho phép các applet nạp vào trình duyệt một cách hiệu quả.
- Một file kê khai (manifest file) chứa thông tin về các file lưu trữ.
- Chữ ký điện tử là một mã hoá kèm với chương trình để nhận diện chính xác nơi nguồn gốc của file.
- Keystore là một cơ sở dữ liệu của các khoá.
- Keytool là công cụ khoá bảo mật của java.
- chứng nhận điện tử là một file, hoặc một đối tượng, hoặc một thông báo được ký danh bởi quyền chứng nhận (certificate authority)

Kiểm tra kiến thức:

1. File _____là file lưu trữ được nén.
2. Tuỳ chọn _____, khi dùng với công cụ jar, trích rút tên file từ một lưu trữ (file)
3. JAR tự động tạo file kê khai, thậm chí nó không được chỉ ra ***true/false***
4. Thuộc tính_____,khi dùng trong thẻ applet, chỉ cho trình duyệt nạp file jar lưu trữ cụ thể, và tìm file class được nhập vào.
5. Trong chữ ký điện tử, _____được dùng cho mã hoá và _____được dùng cho giải mã.
6. Tất cả các thông tin keytool quản lý, được lưu trữ trong một cơ sở dữ liệu gọi là _____
7. keytool ở bước đầu tiên cần nhận được một chứng nhận ***true/false***
8. Gói _____chứa giao diện (interfaces) dùng cài đặt các chính sách điều kiển truy cập.

Bài tập:

Tạo **các câu lệnh java** thực hiện các hành động sau:

1. Tạo một file jar ‘core-java.jar’ chứa các file lớp (class file) và các file nguồn.
2. Liệt kê nội dung của file jar.
3. Tạo file html cho file CardLayoutDemo.class,file lớp được chứa trong file jar.
4. trích rút (extract) file jar
5. Dùng lệnh keytool với tên bí danh và keystore để tạo ra cặp khoá công cộng/riêng mới
6. Ký danh file jar mới được tạo
7. Xác minh chữ ký (signature).
8. Xuất các chứng nhận (certificate)

9. Liệt kê nội dung của keystore
10. In các chứng nhận được lưu trong file.