CHƯƠNG 9
QUẢN LÝ TRUY CẬP: CHỨNG THỰC & CẤP QUYỀN

ThS. Trần Bá Nhiệm
Website:
sites.google.com/site/tranbanhiem
Email: tranbanhiem@gmail.com

Nội dung

- Giới thiệu
- Các kỹ thuật chứng thực
- Chứng thực Microsoft .NET Passport
- Hashing - Băm
- SSL
- Chứng chỉ - Certificates
- Server certificates
- Client certificates
- Cấp quyền trong .NET
- Bảo mật mạng doanh nghiệp

Giới thiệu

- Cho đến hiện nay, chúng ta vẫn thừa nhận hacker dùng phần mềm nghe trộm dữ liệu, nguy hiểm hơn nữa là giả mạo hoặc đánh lừa
- Chương này trình bày những vấn đề phức tạp về vấn đề xác nhận người dùng
- Các hệ thống chứng thực phải có khả năng kiểm tra hợp lệ chứng chỉ và thông điệp không bị làm giả trước, trong, sau khi đến

Giới thiệu

- Chương được trình bày thành 4 phần:
  - Các hệ thống chứng thực Microsoft như: NTLM và .NET Passport
  - Các kỹ thuật phát hiện giả mạo
  - Cơ chế chứng thực SSL cho dữ liệu Web
  - Một số cơ chế chứng thực có liên quan: phân quyền .NET và thừa kế chứng thực

Các kỹ thuật chứng thực

- Để bảo đảm xác minh 1 client, ta cần phải tin cậy vào một mảnh thông tin là duy nhất xác định client đó và chúng không thể dễ dàng xác định hoặc giả mạo (ví dụ: IP, Windows username/password, hoặc một số chứng chỉ khác)
- Các hệ thống chứng thực ngăn chặn giả mạo chứng chỉ nhưng không thể bảo vệ người dùng thiếu cẩn mật

Các kỹ thuật chứng thực

- Với mỗi ngữ cảnh sẽ có một số kiểu chứng thực khác nhau.
- Nếu ta phát triển một giải pháp cho 1 ISP thì ISP có thể xác định chính xác client nào dựa trên địa chỉ IP và như vậy dùng IP làm chứng chỉ.
- Khi phát triển ứng dụng intranet trên Windows ta có thể tin cậy vào quá trình đăng nhập Windows
- Với các dịch vụ Internet có thể dùng tổ hợp tùy chọn chứng thực IIS hoặc username/ password

Các kỹ thuật chứng thực

- Dạng cơ bản và phổ biến là chứng thực bằng cách kiểm tra hợp lệ IP, cho truy xuất thông tin nếu IP thuộc vùng nào đó
- Cơ chế trên được các ISP áp dụng
- IP spoofing (giả mạo IP) có thể làm thất bại kiểu chứng thực này, nhưng cũng không phải dễ dàng

Chứng thực IIS

- Mặc dù chúng ta tập trung vào các phần mềm độc lập, tuy nhiên IIS luôn luôn là một lựa chọn tốt
- Dùng IIS sẽ giúp loại bớt các vấn đề phức tạp, nhất là được dùng các cơ chế mã hóa và chứng thực do Microsoft cung cấp
- IIS5 cung cấp 5 loại chứng thực: Anonymous, Basic, NT challenge/response (NTLM), Integrated Windows (Kerberos), Digest

Chứng thực IIS

- Dạng cơ bản của chứng thực IIS là Anonymous. Client không có bất kỳ chứng chỉ nào và được tự động cấp quyền IUSR (guest): đọc và ghi file
- Basic: bắt buộc client phải cung cấp chứng chỉ ở dạng văn bản thô. Nhược điểm: độ bảo mật thấp. Tuy nhiên nếu kết hợp với SSL thì đây là một giải pháp tương đối tốt

Chứng thực IIS

- NTLM khá an toàn và không thể bẻ khóa nếu không có nỗ lực đáng kể
- NTLM bởi một vài nhân tố xác định
- NTLM được hỗ trợ trong IIS4 và tất cả các phiên bản Internet Explorer
- Chứng chỉ cung cấp bởi client sẽ tương ứng với một tài khoản cục bộ trên server

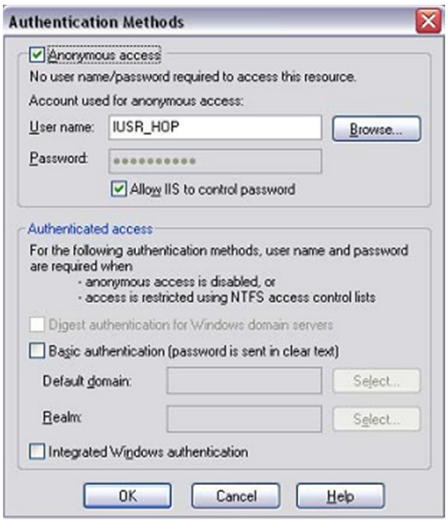
Chứng thực IIS

- Chứng thực kiểu Digest được giới thiệu từ IIS5
- Chưa thấy có công bố nào bẻ khóa được kiểu mã hóa này
- Tương thích với phần lớn phiên bản Internet Explorer
- Chứng chỉ cung cấp bởi client sẽ tương ứng với một tài khoản cục bộ trên server

Chứng thực IIS

- Kerberos cung cấp mức độ bảo mật cao nhất cho chứng thực thông qua Internet
- Yêu cầu truy cập vào domain controller
- Chỉ làm việc trên IIS5 và các phiên bản gần đây của Internet Explorer
- Tinh chỉnh các lựa chọn: Start→ControlPanel→Administrative Tools→Internet Information Services.

Chứng thực IIS



Chứng thực IIS

- Khi chấp nhận các kết nối anonymous, máy tính quản lý 860 request/s, với Basic là 780 request/s - cơ chế chứng thực nhanh nhất mặc dù bảo mật thấp.
- NTLM chỉ còn 99 request/s
- Digest còn 96 request/s
- Kerberos còn 55 request/s
- Với SSL chỉ là 2 request/s

Microsoft .NET Passport

- Chứng thực Passport được dùng với người có thể định danh chính họ nhờ địa chỉ Hotmail.
- Dạng chứng thực này không có ý nghĩa đối với truyền thông thương mại, nhưng rất tốt cho cá nhân
- Thuận lợi trên các hệ thống phục vụ cho gia đình

Microsoft .NET Passport

- Chứng thực Passport có 2 biến thể:
  - Preproduction: miễn phí, nhưng chỉ có một lượng giới hạn thông tin cá nhân khai thác được từ Passport
  - Production: khắc phục nhược điểm trên của Preproduction

Hashing

- Hashing là giải thuật 1 chiều trong đó dữ liệu có thể được băm thành giá trị nhưng giá trị băm không thể chuyển ngược thành dữ liệu ban đầu
- Được dùng kết hợp với mã hóa để bảo đảm thông điệp không bị làm giả trên đường truyền
- Hệ thống hashing hiện đại gồm: Message Digest (MD5) và Secure Hash Algorithm (SHA-1).

Hashing

- Khi một giá trị băm được sinh ra từ khối văn bản gốc, rất khó để tính toán sinh ra một khối văn bản khác cũng có giá trị băm đó
- Đặc tính quan trọng của giải thuật băm là một thay đổi nhỏ trong văn bản đầu vào cũng tạo ra thay đổi rất lớn với giá trị băm

Hashing

- Các giải thuật băm luôn luôn sinh ra các giá trị có cùng độ dài bất kể số lượng văn bản đầu vào
- Trong ứng dụng, một giá trị băm được sinh ra từ thông điệp đã cho và sau đó thông điệp cùng với mã băm được mã hóa với nhau

Hashing

- Khi giải mã thông điệp sinh ra một giá trị băm phải trùng với thông điệp đó, còn ngược lại có nghĩa là thông điệp đã bị làm giả
- Ứng dụng khác nữa đó là cách lưu trữ an toàn username / password. Nếu lưu trữ ở văn bản gốc thì hacker có thể đột nhập và khai thác thông tin này, nhưng nếu đã băm thì họ không thể biết được

Hashing

- Khi người dùng hợp lệ đăng nhập và gõ vào password, nếu password đó sinh ra giá trị băm trùng với dữ liệu lưu trữ thì đăng nhập hợp lệ được thực hiện
- Vấn đề: người dùng quên password?
- Cách giải quyết: tài khoản administrator có thể thiết lập lại các password cho người dùng.

Hashing

- Vấn đề: nếu hacker biết được giải thuật băm?
- Hacker tạo ra password đã băm, thay thế vào một tài khoản đã có và chiếm quyền truy xuất
- Khắc phục: giải thuật băm sẽ được mã hóa với cơ chế khác như 3DES

Hashing

- Băm cũng được dùng để chống khai phá dữ liệu bất hợp pháp đối với các dịch vụ trực tuyến
- Hacker dùng công cụ nghe lén để xác định dữ liệu nào được gửi lên server và tạo ra một sản phẩm dùng dịch vụ đó mà không phải trả phí
- Giải pháp trước đây là mã hóa bất đối xứng, nhưng tổng chi phí cao

Hashing

- Giải pháp mới đề xuất là dùng giá trị khóa băm chèn trong phần header cho tổng chi phí thấp, nhưng không thể tái tạo header được nếu không biết khóa băm đó
- Một số công cụ như Dotfuscator (www.preemptive.com) có thể được dùng để che dấu khóa này

Hashing

- Một ví dụ thực tế của hệ thống này là Google toolbar – hiển thị xếp hạng của trang khi thực hiện tìm kiếm
- Google không muốn nhiều người có thể khai thác những giá trị này thông qua các tiến trình tự động

Các giải thuật Hashing

- .NET cung cấp 2 giải thuật: SHA và MD5 tương ứng trong lớp SHA1Managed và MD5CryptoServiceProvider
- SHA được đặc tả bởi secure hash standard (SHS)
- Khóa băm được sinh ra từ các khối 64 byte
- SHA được định nghĩa trong RFC 3174

Các giải thuật Hashing

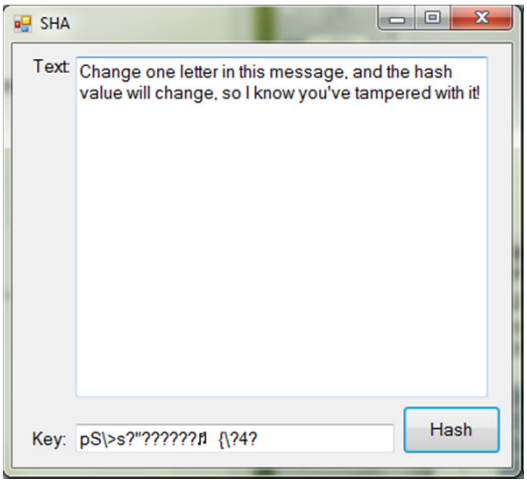
- Tính năng tương tự hàm băm được dùng trong thực tế là cyclic redundancy check (CRC)
- CRC cung cấp một checksum độ dài cố định đối với dữ liệu đầu vào bất kỳ
- 4 biến thể của SHA: SHA1Managed (20-byte hash), SHA256Managed (32-byte hash), SHA384Managed (48-byte hash), SHA512Managed (64-byte hash)

Sử dụng SHA

- Tạo ứng dụng với 1 form, 2 textbox (tbPlaintext, tbHashed), 1 button (btnHash)
- Xử lý sự kiện Click của button:

```
private void btnHash_Click(object sender,
System.EventArgs e)
{
  byte[]  entered =
  Encoding.ASCII.GetBytes(tbPlaintext.Text);
  byte [] hash = new
SHA1Managed().ComputeHash(entered);
  tbHashed.Text = Encoding.ASCII.GetString(hash);
}
```

Sử dụng SHA



SSL

- SSL được định nghĩa trong RFC 2660.
- SSL là dạng bảo mật phổ biến trên Internet mức socket.
- SSL bảo mật giao thức stream, dùng cả mã hóa đối xứng và bất đối xứng, kết hợp với chứng chỉ số để chứng thực
- Chứng chỉ số có thể mua từ các CA như Thawte, Verisign

SSL

- Chứng chỉ chứa thông tin chi tiết về DNS và tổ chức đã mua, được mã hóa với khóa riêng của CA
- Khóa chung của mọi CA được cài đặt vào trình duyệt, vì thế bất kỳ ai trên Internet cũng có thể chắc chắn họ chứ không phải người khác điều khiển máy phục vụ trang mà họ yêu cầu
- Tất cả dữ liệu gửi giữa client và server được mã hóa với RSA

Chứng chỉ

- SSL cung cấp mã hóa và chứng thực tại các thiết bị đầu cuối
- Khi trình duyệt xem một website được bảo mật, một biểu tượng khóa xuất hiện trên thanh trạng thái. Click vào biểu tượng này sẽ xác thực server phụ thuộc vào công ty và vị trí nhất định
- Điều đó đạt được nhờ dùng các chứng chỉ server

Chứng chỉ

- Chứng chỉ được phát hành bởi CA đã được công nhận trên toàn cầu
- Chúng ta có thể tạo chứng chỉ nhưng chỉ được tin tưởng trong phạm vi nội bộ
- Dạng phổ biến của chứng chỉ số là X.509. đây là chuẩn quốc tế được điều hành bởi IETF Public Key Infrastructure (PKIX)
- X.509 có ba phiên bản: v1, v2, v3

Chứng chỉ

- X.509 v3 thường được dùng nhất
- Chứng chỉ bao gồm các trường:
  - Serial number: duy nhất tương ứng mỗi chứng chỉ do tổ chức đó phát hành
  - Signature
  - Validity period
  - Subject
  - Public key
  - Signed hash

Chứng chỉ: Subject

Subject	Ý nghĩa
C	Country
SP	State/province
S	State
L	Locality
O	Organization
OU	Organizational unit
CN	Common name
E	Email

Chứng chỉ server

- Một công cụ hữu ích để tự tạo chứng chỉ server là IBM KeyMan (www.alphaworks.ibm.com/tech/keyman)
- Cũng có thể dùng Keytool nằm trong bộ Java SDK

Chứng chỉ client

- Chứng chỉ server chứng thực 1 website với trình duyệt, ngược lại chứng chỉ client chứng thực trình duyệt với server
- Chứng chỉ client chỉ dùng khi muốn tối ưu hóa bảo mật website như trong dịch vụ ngân hàng trực tuyến
- Chứng chỉ client được cho dùng miễn phí từ Thawte

Chứng chỉ client

- Chứng chỉ client được dùng để gửi và nhận email đã được mã hóa và chứng thực địa chỉ email của bạn với người nhận
- Chứng chỉ client cơ sở chỉ chứng thực địa chỉ email chứ không phải người gửi email
- Xem chứng chỉ trên IE: Tools→Internet Options→Content→Certificates

Các dịch vụ Microsoft Certificate

- Một số tổ chức có thể cần bảo mật nội bộ, như thế thì khá tốn kém khi bỏ tiền mua chứng chỉ cho mọi server, khi đó Microsoft Certificate Services (MSCS) là giải pháp được lựa chọn
- MSCS có thể sinh ra các chứng chỉ X.509
- MSCS có thể hoạt động như một root CA hoặc subordinate CA

Đọc các chứng chỉ

- Dùng các phương thức / thuộc tính của lớp X509Certificate

Phương thức / Thuộc tính	Mô tả
GetCertHashString	Trả về giá trị băm của chứng chỉ ở dạng chuỗi thập lục phân
GetEffectiveDateString	Trả về ngày hiệu lực của chứng chỉ
GetExpirationDateString	Trả về ngày hết hạn của chứng chỉ
GetFormat	Trả về tên của format của chứng chỉ
GetIssuerName	Trả về tên của công ty phát hành chứng chỉ

Đọc các chứng chỉ

Phương thức / Thuộc tính	Mô tả
GetKeyAlgorithm	Trả về thông tin giải thuật khóa
GetKeyAlgorithmParameters	Trả về các tham số giải thuật khóa
GetName	Trả về tên nguyên tắc dựa trên đó chứng chỉ đã phát hành
GetPublicKeyString	Trả về khóa chung của chứng chỉ
GetRawCertDataString	Trả về dữ liệu thô cho toàn bộ chứng chỉ
GetSerialNumberString	Trả về serial number của chứng chỉ

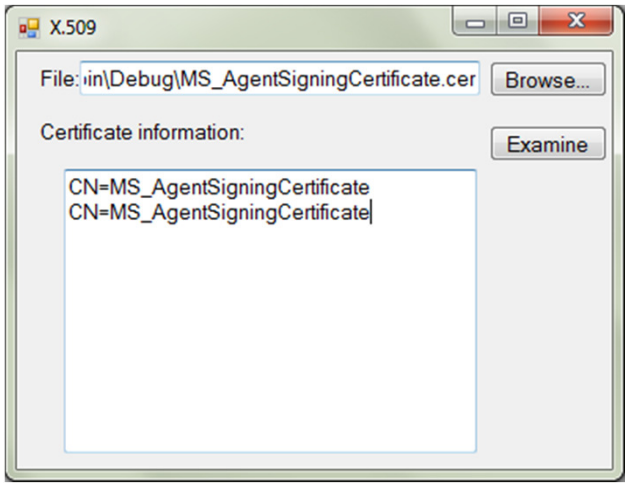
Minh họa

- Tạo project mới
- Thêm 1 form, 2 textbox: tbCertFile, tbDetails, 2 button: btnBrowse, btnExamine.
- Thêm File Open Dialog control
- Dùng đối tượng của lớp X.509certificate để giải mã file và khai thác một số thông tin thích hợp

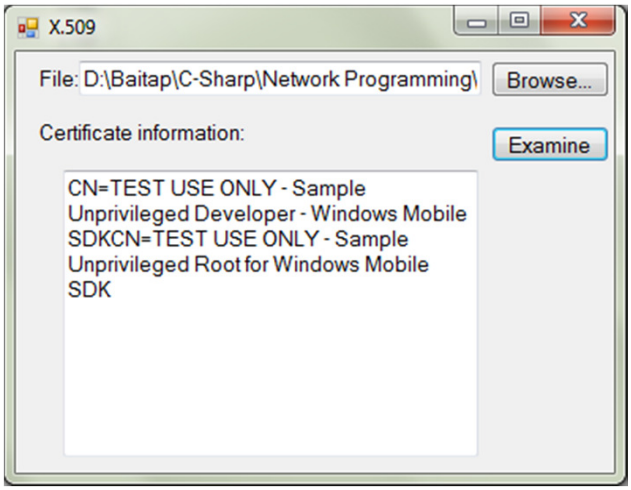
Minh họa

```
private void btnExamine_Click(object sender,
System.EventArgs e)
{
  X509Certificate x509 =
X509Certificate.CreateFromCertFile(tbCertFile.
Text);
  tbDetails.Text = x509.GetName();
  tbDetails.Text += x509.GetIssuerName();
}
```


Minh họa



Minh họa



Phân quyền với .NET

- .NET đưa ra một số kiến trúc sandbox (hộp cát) cho phép người dùng thực thi các đoạn mã không đáng tin cậy mà không sợ ảnh hưởng đến máy tính
- Hiện tại, phần lớn ảnh hưởng đến hộp cát là khi các chương trình thực thi trực tiếp trên thư mục chia sẻ

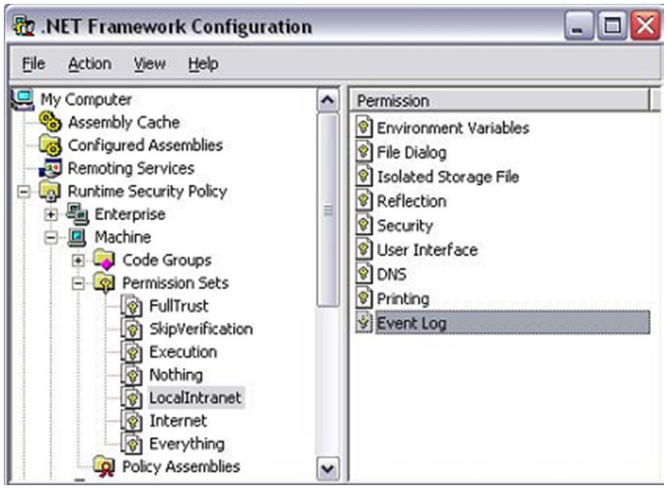
Phân quyền với .NET

- Có một số cách giới hạn thực thi mã chương trình trên thư mục chia sẻ
- Chú ý các đoạn mã không viết bằng ngôn ngữ C# được gọi là unmanaged code
- Bất kỳ hệ điều hành ảo nào trong hộp cát cũng không thể gọi đến unmanaged code
- Giới hạn trên cũng áp dụng cho việc đọc các biến môi trường, truy xuất file log,…

Phân quyền với .NET

- Xem và sửa chính sách an toàn run-time trong .NET, theo thao tác sau: Control Panel→Administrative Tools→Microsoft .NET Framework Configuration, click vào Runtime Security Policy (xem hình)
- System.Security.Permissions namespace cung cấp các công cụ cho phép kiểm tra phân quyền

Phân quyền với .NET



Phân quyền với .NET

- Một đặc tính thú vị trong an toàn truy xuất mã trong .NET là lưu trữ độc lập → đây là ý tưởng không mô phỏng Java
- Đặc tính này cho phép các ứng dụng phát triển trên môi trường intranet hoặc nguồn thứ cấp khác được phép đọc, ghi số lượng dữ liệu có giới hạn vào các máy tính chủ

Phân quyền với .NET

- Nếu ứng dụng có thể đọc và ghi độc quyền, chúng có thể lợi dụng để khai thác các thông tin cá nhân khác như địa chỉ email, nhưng lưu trữ độc lập là giải pháp thông minh để khắc phục vấn đề trên
- Lưu trữ độc lập là vùng không gian đĩa cứng nhỏ (10KB) được cấp phát cho bất kỳ ứng dụng nào đó xuất phát từ nguồn Internet đã tin cậy

Phân quyền với .NET

- Thư mục mà dữ liệu này đặt vào nên tránh các thư mục hệ thống và chứa dữ liệu cá nhân
- Mỗi ứng dụng được cấp thư mục và không gian riêng giống như trường hợp các ứng dụng không tin cậy không thể đọc được dữ liệu của nhau
- Lượng không gian cấp phát có thể điều chỉnh được

Phân quyền với .NET

- Để dùng không gian lưu trữ độc lập, khai báo đối tượng thuộc lớp IsolatedStorageFile và tạo đối tượng stream gắn vào nó
- Stream trên sẽ được dùng tương tự như FileStream

Phân quyền với .NET

```
IsolatedStorageFile IsolatedStore;
IsolatedStorageFileStream IsolatedStream;
IsolatedStore =
IsolatedStorageFile.GetStore(IsolatedStorag
eScope.Assembly, null,null);
IsolatedStream = new
IsolatedStorageFileStream("data.txt",
  FileMode.CreateNew, IsolatedStore);
```

An toàn mạng thương mại

- Nếu hacker xâm nhập thành công vào website thương mại và chiếm một số thẻ tín dụng thì chỉ có một số người không may mắn sở hữu thẻ đó bị thiệt hại
- Nếu điều tương tự xảy ra với ngân hàng Nhà nước hay thế giới thì nghiêm trọng hơn rất nhiều – nền kinh tế của quốc gia có thể phá sản sau 1 đêm

An toàn mạng thương mại

- Phần lớn ngân hàng dùng các đường thuê bao riêng giữa các chi nhánh vì thế thông tin bí mật không giao tiếp trên mạng điện thoại công cộng
- Các máy ATM dùng các kết nối VPN đến ngân hàng
- Các ATM bị giới hạn số lượng giao dịch tối đa có thể thực hiện, vì vậy không thể dùng để tấn công ngân hàng

An toàn mạng thương mại

- Khi ngân hàng cần truyền thông với tổ chức thương mại khác, họ phải dùng mạng điện thoại công cộng. Nếu giao dịch diễn ra hàng ngày, phải thiết lập private virtual circuit (PVC)
- Truyền thông này được mã hóa rất mạnh với ISO 8730 hoặc SWIFT

X.25

- Rất nhiều dịch vụ thương mại điện tử chạy trên giao thức X.25 chứ không phải IP
- X.25 được phát triển bởi CCITT vào năm 1978 và được dùng phổ biến trên các mạng ngân hàng
- X.25 hỗ trợ rất nhiều đặc tính của TCP/IP như hướng kết nối và tính nhất quán dữ liệu nhờ high-level data link control/Link access procedure balanced (HDLC/LAPB)

X.25

- X.25 được phát triển bởi CCITT vào năm 1978 và được dùng phổ biến trên các mạng ngân hàng
- X.25 hỗ trợ rất nhiều đặc tính của TCP/IP như hướng kết nối và tính nhất quán dữ liệu nhờ high-level data link control/Link access procedure balanced (HDLC/LAPB)

X.25

- X.25 hỗ trợ cả công nghệ mạch ảo và PVC
- Số mạch ảo có thể hỗ trợ tối đa là 200/mỗi đường X.25
- Trường hợp dùng X.25 trên mạng IP thì LAPB có thể thay thế TCP/IP. Phần mềm Cisco IOS hoặc các TCP X.25 gateway có khả năng tương tự (xem RFC 1613)

ISO 8730

- Mặc dù ít phổ biến hơn SWIFT nhưng dạng này thường dùng cho truyền thông giữa các ngân hàng trên thế giới
- Dùng khóa đối xứng với cơ chế phân phối khóa ISO 8732 / ANSI X9.17. Trung tâm phân phối khóa có thể chạy tại một hoặc nhiều ngân hàng khác nhau, hoặc do bên thứ ba được tin cậy

ISO 8730

- Mỗi thông điệp ISO 8730 có thể được băm bằng một trong hai cách:
  - Toàn bộ thông điệp
  - Chỉ những chi tiết chủ yếu
- Mỗi thông điệp phải chứa ngày được địa chỉ MAC tạo lập → hủy các thông điệp quá hạn
- Giá trị ngày cũng được băm tùy theo chế độ hoạt động

ISO 8730

- Các trường thông tin được băm và được phân tách rõ ràng gồm:
  - QD<date>DQ: ngày được MAC tạo ra
  - QK<key>QK: khóa chứng thực được dùng bởi client
  - QX<message ID>XQ: số duy nhất cho ngày và khóa trên
  - QT<transaction detail>TQ: chi tiết của số giao dịch
  - MQ<hash>MQ: băm chính nó, dài 8 byte

SWIFT

- Society for Worldwide Interbank Financial Telecommunications (SWIFT)
- Mạng SWIFT phục vụ cho trên 7000 tổ chức thương mại ở 200 quốc gia
- Để truy cập được vào mạng trên cần phải có thiết bị đầu cuối đặc biệt, cùng với phần mềm được chứng nhận bởi SWIFT

SWIFT

- Truyền thông trên mạng dùng X.25 hoặc Secure IP Network (SIPN)
- Kết nối vào SWIFT với đường thuê bao riêng hoặc ISDN độc quyền
- Một API được SWIFT cung cấp để truyền thông với mạng này

Bài tập

- Cài đặt các chương trình đã minh họa trong bài giảng của chương bằng ngôn ngữ C# hoặc VB.NET