6

Lab

Bảo vệ dữ liệu trong Truyền thông với C#

Data Protection with C#

Thực hành Lập trình mạng căn bản

GVHD: Phan Trung Phát

Lưu hành nội bộ

A. TỔNG QUAN

1. Mục tiêu

- Nắm được các kiến thức về mã hóa, các phương pháp mã hóa.
- Áp dụng mã hóa nhằm bảo vệ dữ liệu trong quá trình truyền thông qua mạng.

2. Môi trường

- IDE Microsoft Visual Studio 2010 trở lên.

3. Liên quan

- Sinh viên cần nắm được các kiến thức nền tảng về lập trình. Các kiến thức này đã được giới thiệu trong các môn học trước và trong nội dung lý thuyết đã học.
- Các kiến thức nền tảng về bảo mật, mã hóa, các thuật toán mã hóa.
- Tham khảo tài liệu (Mục D).

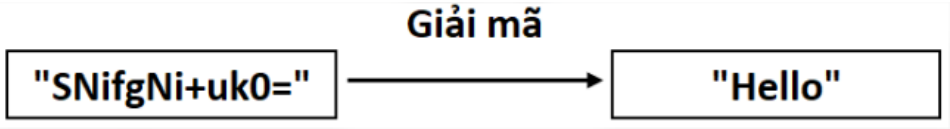
B. THỰC HÀNH

1. Cơ bản về mã hóa dữ liệu

Khi truyền qua môi trường mạng, dữ liệu có nguy cơ bị rò rỉ, đánh cắp. Một trong các phương pháp để bảo vệ dữ liệu, được sử dụng phổ biến là mã hóa.

Mã hóa là quá trình chuyển dữ liệu từ dạng này sang dạng khác nhằm bảo vệ tính bí mật. Thông thường đó là quá trình chuyển dữ liệu từ dạng có thể đọc hiểu được (bản rõ – plaintext) sang dạng không thể đọc hiểu bằng cách thông thường (bản mã – ciphertext). Thực tế, việc mã hóa dữ liệu sẽ không thể nào ngăn việc dữ liệu có thể bị đánh cắp, nhưng nó sẽ ngăn việc người khác có thể đọc hiểu được nội dung của dữ liệu.

Giải mã là quá trình ngược lại của mã hóa, tìm lại bản rõ (plaintext) từ bản mã (ciphertext).



Quá trình mã hóa hoặc giải mã thường đi kèm khóa (key), dùng để mã hóa hoặc giải mã dữ liệu.

2. Các phương pháp mã hóa

2.1. Mã hóa cổ điển

Mã hóa cổ điển là các phương pháp mã hóa có tính đơn giản, thường áp dụng kỹ thuật mã hóa dựa trên thay thế kí tự. Do đó, loại mã hóa này không được sử dụng phổ biến vì dễ dàng phá mã thông qua vét cạn.

Một số phương pháp mã hóa cổ điển:

- Mã hóa Caesar
- Mã hóa Vigenère
- Mã hóa Playfair

Caesar

Đây là một trong những phương pháp mã hóa dạng thay thế đơn giản nhất, được sử dụng bởi Julius Caesar để giao tiếp với quân đội của ông. Trong phương pháp này, mỗi ký tự trong bản rõ được dịch sang phải n lần để có được ký tự mã hóa tương ứng trong bản mã.

Ví dụ: bản rõ "ILOVEUIT", khi được mã hóa với n = 2 thì có được bản mã "KNQXGWKV".

Vigenère

Mã hóa Vigenère là phương pháp mã hóa sử dụng xen kẽ vài phép mã hóa Caesar với các bước dịch khác nhau. Trong phương pháp có một ma trận hình vuông Vigenère gồm 26 hàng, mỗi hàng dịch về bên trái một bước so với hàng phía trên, tạo thành 26 bảng mã Caesar.

Phương pháp này chọn một khóa, được viết lặp lại nhiều lần trên một dòng cho đến khi số ký tự bằng với số ký tự của bản rõ. Sau đó, từng cặp ký tự cùng vị trí trong bản rõ và từ khóa sẽ được dùng làm cột và dòng tương ứng trong ma trận vuông Vigenère để tìm ký tự tương ứng trong bản mã.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

Hình 1. Ma trận hình vuông Vigenère

Ví dụ: bản rõ ILOVEUIT, từ khóa LEMON sẽ được viết thành LEMONLEM.

Bản rõ: I L O V E U I T

Khóa: L E M O N L E M

Bản mã: T P A J R F M F

2.2. Mã hóa đối xứng

Mã hóa đối xứng là phương pháp mã hóa chỉ sử dụng một khóa cho cả quá trình mã hóa và giải mã. Các thuật toán mã hóa đối xứng mã hóa dữ liệu trong các khối, thường đệm thêm dữ liệu để đảm bảo các kích thước khối giống nhau được sử dụng cho mỗi khối. Mỗi khối dữ liệu được mã hóa được liên kết với nhau.

Một số phương pháp mã hóa đối xứng:

- DES (Data Encryption Standard)
- 3DES (Triple Data Encryption Standard)
- AES (Advanced Encryption Standard)

DES và 3DES

DES (Data Encryption Standard) là một thuật toán khối với kích thước khối 64 bit và kích thước khóa 56 bit, được Tổ chức Tiêu chuẩn xử lý thông tin liên bang Hoa Kỳ (FIPS) công bố chính thức vào tháng 11/1976.

3DES (Triple Data Encryption Standard) là tên chung của việc áp dụng mã hóa một khối dữ liệu qua 03 lần mã hóa DES với 03 khóa khác nhau.

AES

AES (Advanced Encryption Standard) được Viện Tiêu chuẩn và Công nghệ quốc gia Hoa Kỳ giới thiệu vào năm 2001. AES là một thuật toán mã hóa khối đối xứng với độ dài khóa là 128 bit, 192 bit và 256 bit tương ứng dọi là AES-128, AES-192 và AES-256.

2.3. Mã hóa bất đối xứng

Mã hóa bất đối xứng là phương pháp mã hóa sử dụng hai khóa cho quá trình mã hóa và giải mã. Một khóa gọi là khóa công khai (public key) và một khóa là khóa bí mật (private key). Với khóa công khai, tất cả mọi người đều có thể có được khóa này. RSA là một trong những hệ thống mã hoá bất đối xứng được sử dụng rộng rãi trong mã hoá và thiết lập chữ ký điện tử.

2.4. Mã hóa một chiều

Phương pháp mã hóa này để mã hóa những dữ liệu không yêu cầu giải mã lại bản nguyên bản gốc. Thông thường phương pháp mã hóa một chiều sử dụng một hàm băm (hash function) để biến một chuỗi thông tin thành một chuỗi hash có độ

dài nhất định. Không có bất kì cách nào để khôi phục (hay giải mã) chuỗi hash về lại chuỗi thông tin ban đầu.

Các thuật toán mã hóa một chiều (hàm băm) thường gặp nhất là MD5 và SHA.

2.5. Mã hóa dữ liệu trong C# với thư viện hỗ trợ

C# cung cấp namespace ***System.Security.Cryptography*** nhằm hỗ trợ cho các hoạt động mã hóa và giải mã dữ liệu.

Mã hóa đối xứng

Bảng 1. Một số lớp hỗ trợ mã hóa đối xứng

Mã hóa	Lớp
AES	System.Security.Cryptography.Aes
DES	System.Security.Cryptography.DES
RC2	System.Security.Cryptography.RC2
3DES	System.Security.Cryptography.TripleDES

Mã hóa bất đối xứng

Bảng 2. Một số lớp hỗ trợ mã hóa bất đối xứng

Mã hóa	Lớp
Digital Signature Algorithm (DSA)	System.Security.Cryptography.DSA
RSA	System.Security.Cryptography.RSA

Mã hóa một chiều

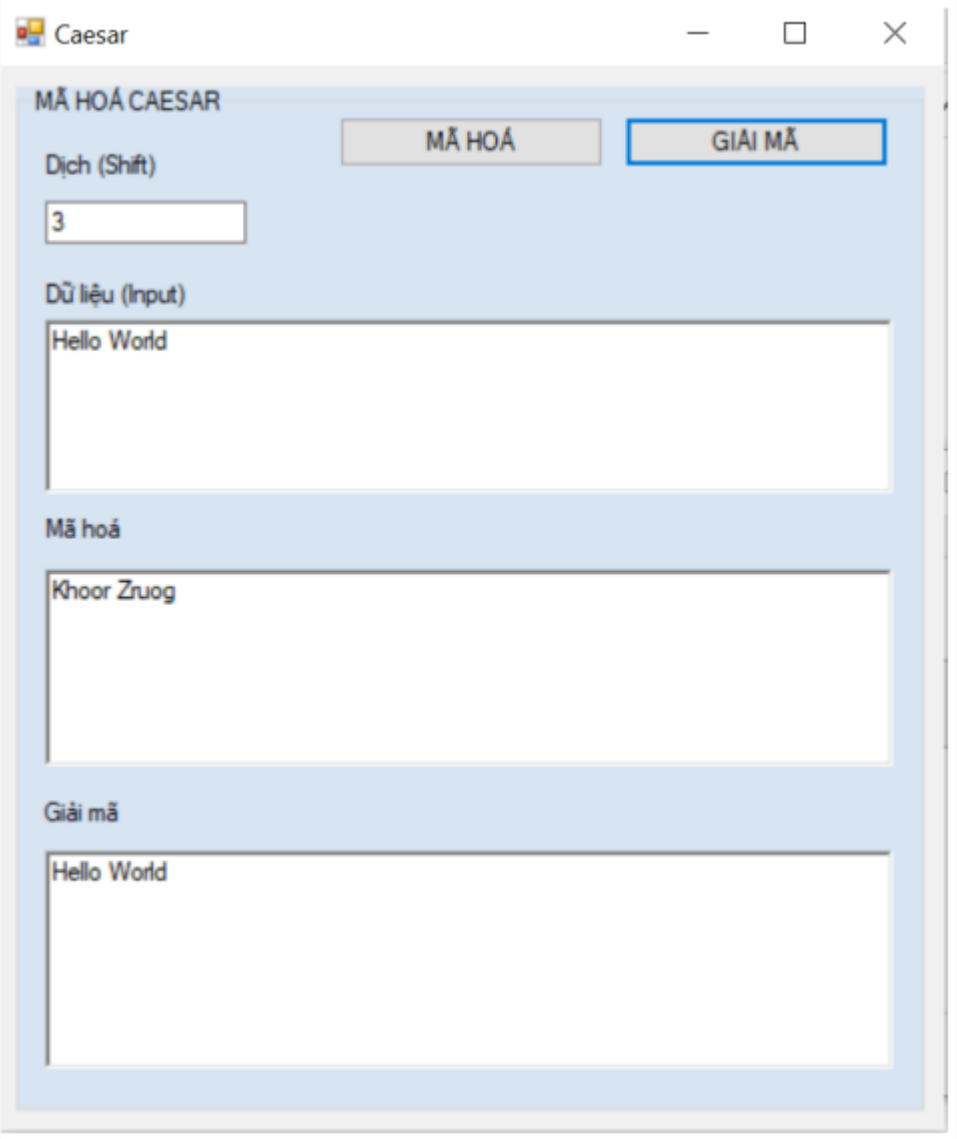
Bảng 3. Một số lớp hỗ trợ mã hóa một chiều

Mã hóa	Lớp
MD5	System.Security.Cryptography.MD5
SHA256	System.Security.Cryptography.SHA256
SHA512	System.Security.Cryptography.SHA512

3. Nội dung thực hành

Bài 1: Viết chương trình cho phép mã hóa và giải mã dữ liệu theo phương pháp mã hóa Caesar.

Giao diện tham khảo:



Bài 2: Tương tự như Bài 1 nhưng thực hiện trên Vigenère.

Lưu ý: Tại Bài 1 và Bài 2, thực hiện đọc và hiểu các nội dung liên quan đến thuật toán mã hóa cổ điển Caesar và Vigenère. Từ đó, thực hiện triển khai lại bằng C# và không sử dụng thư viện được cung cấp sẵn.

Bài 3: Cài đặt một thuật toán mã hóa cho 1 ứng dụng mạng. Sử dụng lại kết quả của bài thực hành số 3, bao gồm: 1 Server và tối thiểu 2 Client. Thực hiện cài đặt 1 thuật toán mã hóa AES hoặc RSA trong việc mã hóa hoặc giải mã dữ liệu được trao đổi từ Client đến Server và ngược lại. Từ đó, giúp dữ liệu được bảo vệ.

Lưu ý: Được phép sử dụng các thư viện được cung cấp sẵn bởi C#.

C. YÊU CẦU & NỘP BÀI

1. Yêu cầu

- Mỗi bài tập là 1 project và đặt trong cùng 1 solution.
- Mỗi project đều có form điều hướng để mở các form liên quan.
- Các giao diện ở trên chỉ mạng tính chất minh họa, sinh viên tiến hành thiết kế giao của riêng mình đảm bảo các tiêu chí: dễ nhìn, thể hiện hết được các yêu cầu cần thực hiện, đẹp.
- Code "sạch" [2], đặt tên biến rõ ràng.
- Nộp bài không đầy đủ; lỗi, không chạy được; nộp trễ; sao chép code bạn khác, nguồn có sẵn: *xử lý tùy theo mức độ.*

2. Đánh giá kết quả

- Sinh viên thực hành và nộp bài theo **Nhóm (Nhóm trưởng nộp)** tại website môn học theo thời gian quy định.
- Bài nộp bao gồm toàn bộ Source-code của các bài tập liên quan tại github và trình bày báo cáo gồm Ảnh chụp màn hình kèm mô tả, giải thích các bước hoạt động, thực hiện của ứng dụng đã viết trong từng bài:

Toàn bộ project đặt vào 1 file nén (.zip) với tên theo quy tắc sau:

Mã lớp-LabX-MSSV1-MSSV2

Ví dụ: *NT106.M21.MMCL.1-Lab06-20520001-20520002*

D. THAM KHẢO

[1] Microsoft (2018). C# Guide. [Online] Available at: https://docs.microsoft.com/en-us/dotnet/csharp/

[2] Martin, R. C. (2009). *Clean code: a handbook of agile software craftsmanship*. Pearson Education.

[3] Network Programming in the .NET Framework: https://docs.microsoft.com/en-us/dotnet/framework/network-programming/

[4] Cryptography: https://learn.microsoft.com/en-us/dotnet/api/system.security.cryptography?view=net-7.0

HẾT