# COM 5336 Final Project

DUE BY 11:59PM **6/17**/2017 (**Sat**)

10% penalty applies to 1-day late submissions received between 0:00AM 6/18(Sun) and 11:59PM 6/18. No submission will be accepted after 0:00AM 6/19/2017 (Mon).

## Objective

Implement the XTR-Diffie-Hellman Key Exchange.

## Description

In order to implement this assignment, you need to implement the corresponding units one by one. (cf. XTR original paper.)

1. GF($p$): addition, negation, multiplication, inverse. (Already done in previous assignments)
2. GF($p^2$): addition, negation, multiplication, inverse, $p$-th power. (normal basis representation)
3. $S_n(c)$ computation subroutine (Algorithm 2.37).

For simplicity, your plaintext is already assumed to be an element in GF($p^2$), represented as a pair of GF($p$) elements. Therefore you don't need to worry about conversions. You can use any existing code in any previous assignments. Your program must be able to handle at least **160-bit *q*** and **170-bit *p***.

## Test Cases (3 cases, input shown in bold face)

```
==========================================================================
CASE1
==========================================================================

p = 0ED63D1E 55533501 0A551576 3B2A9D57 DAD4CFE6 2E98A34B
q = 0B507397 0D0FEDC5 4F45BA84 36122A02 29C0CF9F B4F6D7
Tr(g) = ( 05979DD0 F29B6D57 87458232 E258C35C 12898230 E4610C58 ,
0C328613 C13A98B2 42CEF504 BBFAAE70 15C6C0F4 D3BCF162 )
a = 9166EB14 BA0E8757 A9FBCF6A D13710AE A4EC4CF3 288A
b = 09C4B2B4 734B0E8B 24119D8C 350AC19F 199FDEB2 BA5DA6


Tr(g^a) = ( 02C5E47C A6F54961 1E8AFF5E 1FD9774A DC595F96 C423B05F ,
02AF7114 8BF8B99C 70443E40 6D25BEBA 51B649EF C8F15B09 )
Tr(g^b) = ( 09D4B23A 30472600 9A7ED23D 399F03AE 06249BD5 DBAE4EB8 ,
0EBF9B6E 2684C98D 7E197AC1 4EF5B706 07E9CAD6 515ACB83 )
Tr(g^ab) = ( 0A1C846D 71723116 87581025 4D5AB395 76197AC2 3AE0B176 ,
0870082D 9DB05C2F 6052BBC5 2B144D16 7C5CB5FF 1FB13D62 )




==========================================================================
CASE2
==========================================================================
p = 34E3B314 F5E9A472 33F5B4FB 0A2581CB 2C6F0A56 9C42E7
q = 0108D7C8 597803B7 090595B4 081CC1BA 06AAAE4C 919943
Tr(g) = ( 0D88D13A 37D809A3 C4F56634 7DC25A19 76E15AF2 5586EC ,
19CA3FCA 322A8526 195FB434 181F2EC2 A6B51252 6AD550 )
a = 85AD6CF2 845F6C3F AA8C6F3E 846328FD 123D
b = 9166EB14 BA0E8757 A9FBCF6A D13710AE A4EC4CF3 288A


Tr(g^a) = ( 00223503 D4EAFFB7 26200FEC 8D1B2A1B 44979106 1B47661D ,
00205653 2E9BB60E 77B8DAA4 46235ECA 26569CBB A4F041E1 )
Tr(g^b) = ( 0027C7A6 112A821D AE6E9CFA 4166F95E E6A0D9DA 05976FF0 ,
```

```
002FBD74 89FB81C8 87F52A27 ACA78807 49739C36 1043D00B )
Tr(g^ab) = ( 00067F77 852D8C03 6443BECD BC1D4768 5F22B5B6 CD25F126 ,
00212015 A9A56506 05D45758 1161F425 B8FB9C14 D31381FE )




========================================================================
CASE3
========================================================================
p = 0108D1D3 E17C76E2 2E46A155 81BF2133 FB46E9F7 24373411
q = 0440E60D 61F2F12B 12BA3592 2B70B2E6 8495D8CF BAD837
Tr(g) = ( 10228371 6FACBF83 41AF7A37 F58F4BAE F823C228 0C5817 ,
8C3EBF1C 3295F4E8 901CB314 FF406921 962DA40A D2F9AB )
a = 038AD6C3 8A7E9F63 1567AF6D 3E9F641D FDEA3C6D 84621A
b = 02F84498 6C1F6E3E 9A183E5E 9138C9C9 C06D1B67 F74098


Tr(g^a) = ( 000154F0 4116EC28 601B8A16 B20484CF 6F5B8E25 592A41B4 ,
00713953 FB893257 9BAFCBB5 9C006A13 ADD239F4 D872ECB6 )
Tr(g^b) = ( 00902EBE 9A8B7803 D535FA72 EEA20D43 811C5C5D FEFAC755 ,
006A4ADD 490A8A80 9BF8F346 18D2FF7F 32A32EE3 C1A19757 )
Tr(g^ab) = ( 005CE9B7 260F4F6F A86CAA61 6343CF4B D4E3F553 F9E801C7 ,
00D7BE92 EF3E5408 26693C62 EC3CC91C 835516B1 FE922221 )
```

## Grading

Your program MUST BE compatible with Dev C/C++ or GNU C/C++ compilers. If you are using other compilers, please make sure your final program is compatible. You will get no points if your program is not compilable using the abovementioned compilers. If your program is compilable but the result is not completely correct, you'll still get partial credits. Your program should be well-commented, well-structured, and easy to understand. You may lose up to 30% of points if you fail to do so.

## Submission

Put all your source codes in a folder containing main functions, function implementations, class definitions, or compilation instructions (if any). Compress them as a single zip file. DO NOT submit executable files. Name your zip file as your student ID number (i.e. 100012345.zip). Submit your source code on iLMS at http://lms.nthu.edu.tw.