E0 254 August-December 3:1

Network and Distributed Systems Security

Instructor

R. C. Hansdah
Email: hansdah@iisc.ac.in

Teaching Assistant

Email:

Department: Department of Computer Science and Automation

Course Time: Mon-Wed 8:00-9:30 AM

Lecture venue: CSA Multimedia Class Room 252

Detailed Course Page:

Announcements

Brief description of the course

The course is intended for students with a background in computer science and programming at the undergraduate level. The objective of the course is to train the students so that they can design and implement security component of the applications they are dealing with.

Prerequisites

Knowledge of C++ or Java is required.

Syllabus

Security Requirements of Distributed Systems; Security Violations, Security Goals, Security Services, Security Protocols, and Security Mechanisms; Attack on Security Protocols and Security Mechanisms; Secret Sharing Techniques and One-Way Functions; Discrete Logs, Block Encryption/Decryption Functions, Hash Functions, and MAC Functions; Algorithmic Implementation and Security Requirements of One-Way Functions; OS Security Violations and Techniques to Prevent Them; Access Control Models; Authenticated Diffie-Hellman Key Establishment Protocols; Group Key Establishment Protocols; Block Ciphers and Stream Ciphers; Block Cipher Modes of Encryption; Nonce, Timestamps and Authentication Protocols; Digital

Signatures and Source Non-Repudiation Protocols; PKI and X.509 Authentication Service; Security Protocol Verification: Strand Space Theory; Kerberos; E-mail Security; Security Issues in Layered Communication Models: IP Security, Secure Socket Layer and Transport Layer Security; Secure Electronic Transactions; Intrusion Detection; Malicious Software Detection; Firewalls.

Course outcomes

At the end of the course, a student of the course is expected to know the following:

1. Knowledge of the mathematical basis for various one way functions, viz., RSA cryptography, prime number based discrete log, ECC based discrete log, block encryption functions, hash functions, MAC functions,pseudorandom number generators, and how they are designed.

2. How these one way functions are used to design a security protocol to meet the security requirements of a distributed system.

3. How to identify the security requirements of a dsitributed systems, and design a security protocol to meet these requirements.

4. The student is able to deal with 512 or 1024 bit integers in C++ or Java and is able implement a security protocol in C++ or Java using 512 or 1024 bit integers.

Grading policy

20% - Two mid term tests.

30% - Three common programming assignments dealing with 512 0r 1024 integers and an individual assignment in which they have to design and implement a security protocol using 512 or 1024 bit integers in a network of PCs.

50% - Final examination.

Assignments

Assignment No. 1

1. Write a program in C++ to find a 1024-bit prime number. The program should output the prime number in

both binary and decimal. You can use the Rabin-Miller or any other algorithm to find the prime number. The program should output the time required to find the number along with the number, and then wait for the instruction to either exit or produce the next prime number.

2. Using the 1024-bit prime number found in (1) above, say n, write a program to find the multiplicative inverse of a given number modulo n using the extended Euclidâ€™s algorithm. In this case, the program should first ask for the prime number (in decimal), and then for the number whose inverse (in decimal) is to be found. The program should then print out the multiplicative inverse (in both binary and decimal), and also give a check for its correctness.

Assignment No. 2

Use output feedback (OFB) mode of block encryption to generate a sequence of bytes using AES block encryption function. Carry out suitable tests (including next-bit test) to find out whether the sequence of bits generated is cryptographically secure. Get this result for at least three pairs of input vector and shared secret key, and three different sizes (128, 192, and 256) of shared secret key (a total of 9 outputs). Now reduce the number of rounds (by reducing the main round function) in the AES block encryption function by half (5 for 128-bit key, 6 for 192-bit key, 7 for 256-bit key), and carry out the same test again using the same pairs of input vector and shared secret key. Then submit the following by e-mail (tarred and compressed files in a directory on or before October 29, 2017). You may use an existing implementation of AES in your program.

1. Source code of the program.
2. Description of each test program used.
3. Results for each test program for each pair of input vector and shared secret key used.
4. Your conclusions from the experiment.

Assignment No. 3

Write a program in Java or C++ to establish 128-bit shared secret key between two processes in different machines with the help of Authenticated Diffie-Hellman key establishment protocol. That is, the processes already have a shared secret key, and they want to establish a new shared secret key ensuring perfect forward secrecy. For authentication, you can use any of the MAC function. The program should use the following two types of groups to establish the shared secret key.

(i) Multiplicative residue group , where p is a 512-bit prime number.
(ii) Elliptic curve group , where p is a 512-bit is a prime number.

For the case (i), you need to get a prime subgroup of , where Diffie-Hellman key establishment protocol can be run. For the case (ii), you can use some standard elliptic curve, where Diffie-Hellman key establishment protocol can be run in a prime subgroup of the main group. Please do not use any library function to compute the exponentiation function. The program should display the common key established in a separate window for each process, and the time required to compute the key (excluding communication delay) in each of the group. Note that this assignment would be required in assignment 4 which is different for each student. Please submit the tar and compressed file of your program, and the output of your program within the due date along with explanation for how to run the program. You will be required to give the demo of your program later.

Note: You may use the attached C++ program as a basis to develop your program. Do not use existing library to find the prime number or inverse. If you do so, you will automatically get zero marks for your assignment. A demo of the programs needs to be given sometime later.

Assignment No. 4

Write a program to achieve the following. Let A, B, C be three nodes. Nodes B and C get a public key certificate about their public key from node A. Then they use the public key certificate to establish a shared key between them. Nodes B and C would give the following information to node A, which would issue the public key certificate to them after verification through other channel like mobile number and e-mail. In the assignment, you do not need to do these verifications. However, note that node A still needs to verify that the sender indeed has the corresponding private key. You can only use 512-bit integer as in case of assignment 3. You can assume that the public key of node A is known.

(i) Name of the user(In reality as recorded for mobile number or Aadhar card or NIN)

(ii) Mobile number

(iii) E-mail address

(iv) Aadhar number or any other national identification number

(v) Diffie-Hellman or RSA public key

(vi) Type of public key

(vii) Public key parameters

(viii) Other functions such as hash function used in the public key certificate

If you use Diffie-Hellman public key, you can use assignment 3 to establish the shared key using the public key certificates. But if you choose to use RSA public key, you need to use a suitable protocol to establish a shared secret key using the public key certificates.

Please submit the tarred and compressed files of your assignment using e-mail along with a description of the assignment. The description should include the protocols used in the implementation, and the traces of output.

Resources

Network of PCs running linux