Misc

• Homework #4 is DUE as of 5PM EST next week – Dec 6th
• ALL HOMEWORK IS DUE on DEC 6th @5PM EST
• Unless previously arranged with me, NO HOMEWORK WILL BE ACCEPTED AFTER THAT
Lecture 11
WWW Security
Who do you trust?

• Who?
  – Family, friends, government, big companies, university, …

• Why trust them?

• Are you sure they are who they say they are?
  – What if I’m not really Professor Poelman, how would you know? Maybe I’m an imposter and have tied the real professor up, somewhere.

• Why are you sure that’s who they are?
  – “vouched for”, proof, ?

• What if they aren’t?
  – Spoofing, sniffing, phishing
Security

- **Identification** – who do you say you are
  - login
- **Authentication** – verifying you are that entity
  - password, id token, fingerprint
- **Authorization** – what can you see and do
  - Access Controls
- **Privacy (Encryption)** – preventing others from seeing the data
  - Encryption
- **Integrity** – proving that the data wasn’t tampered with.
  - Checksums, hashes, signature
- **Non-repudiation** – sent me order and now you are trying to say you didn’t
  - Logging digital signatures and copies of received messages
- **User Access Management** – granting and revoking access privileges and entitlements.
  - CRL – Certificate Revocation Lists
Identification

• Who do you say you are?
  – Login
  – ID Card (ATM card, Smart Card, Drivers License)
  – Speech Recognition
  – Face Recognition (combined with Authentication)
  – Iris Scan (combined with Authentication)
  – ????
Authentication

• Verifying you are who you claim to be
  – You possess a secret
    • Password, some knowledge
    • Id token – ID card, dongle
  – A unique physical characteristic i.e. biometric authentication (also called biometric identification)
    • Fingerprint
    • Hand recognition
    • Iris Scan
    • Voice Print
    • Face Recognition
    • Signature Recognition
    • DNA?
Authorization

• what can you see and do
  – Access to information and application
  – Changing data
  – Running code

• ACL – Access Control Lists
• Permissions
• Entitlements
Privacy (Encryption)

- Preventing others from seeing the data
- Using a mathematical algorithm to transform the data in a way that obscures the content from people observing it, yet the use of another algorithm (or the same) on that obfuscated data results in the original data being outputted.
- Need an algorithm that can’t be easily cracked or reverse engineered by crypto techniques
- Nice if the algorithms were parametric i.e. by providing a key and the original data you can make the encryption even harder to crack and even if you crack that message it doesn’t give you access to future messages.
Encryption

- Simple algorithm – shift each letter by 1 up
  - Ex:
    - A B C D E becomes B C D E F
    - “HELLO” becomes “IFMMP”
    - $E = f(T + X)$ is a basic algorithm where $T$ is the original character, $E$ the encrypted character, $X$ is the number of characters to shift by (the key)
    - $T = f(E - X)$ is the decryption algorithm
    - “Secret Decoder Ring” Algorithm
    - Easy to crack – same character results in same encrypted character – look for spaces, common double characters, common diphthongs, etc.
Symmetric Key
(single key / private key)

- Real encryption
  - A single shared key is used by sender and receiver
  - Key must be kept private or “sniffers” could read your message
  - Example machine: Nazi’s enigma machine WWII
  - Example algorithms: DES, RC4, IDEA
Asymmetric (Public) Key (key pair / public & private keys)

- A pair of keys are generated
  - Public Key - one that is published and obtained by the SENDER
  - Private Key - is retained by the RECEIVER and kept private.
- The sender uses the PUBLIC key to encrypt a message and then sends it to the RECEIVER.
- The RECEIVER used the private key to decrypt the message.
- A better analogy might be that the public key is actually a lockbox posted publicly that the SENDER places the message into (encrypts it) and it can only be unlocked by my private (RECEIVER) key.
- Problems:
  - Slower algorithms than secret key (symmetric key)
  - How do I know who really sent the message was you?
    - Authenticating the Sender
    - Generating and publishing keys
Public Key Analogy

• Locking bank bag (or a lock box I send you with the lock attached and open) is the public key, the key that I have that opens the bag is the private key.
• I give you the bag, unlocked, you place the contents in the bag, lock it, and then send the bag back to me.
• Only I can unlock the bag (open the box) – you can’t
• Questions
• Comments

• 7.5
RSA Algorithm for PK (public key) Encryption

• “Created“ in 1997 by Rivest, Shamir and Adleman at MIT

• Uses large Primes P and Q

• Used to generate two key pairs
  – E & N are the public key pair
  – D & N are the private key pair
  – You publish E & N and keep D private
RSA Algorithm for PK (public key) Encryption (2)

• Public Key is:
  – n, the modulus, a nonnegative integer
    e, the public exponent, a nonnegative integer
  – In a valid RSA public key, the modulus n is a product of two odd primes p and q, and the
    public exponent e is an integer between 3 and
  – n-1 satisfying gcd (e, lambda(n)) = 1, where lambda(n) = lcm (p- 1,q-1).

• Private Key is:
  – n, the modulus, a nonnegative integer
    d, the private exponent, a nonnegative integer
  – The second representation consists of a quintuple (p, q, dP, dQ, qInv), where the
    components have the following meanings: p, the first factor, a nonnegative integer
    q, the second factor, a nonnegative integer
    dP, the first factor's exponent, a nonnegative integer
    dQ, the second factor's exponent, a nonnegative integer
    qInv, the CRT coefficient, a nonnegative integer
  – the modulus n is the same as in the corresponding public key and is the product of two odd
    primes p and q, and the private exponent d is a positive

– http://www.cis.ohio-state.edu/cgi-bin/rfc/rfc2437.html
PK Encryption

- I generate the key pairs
- I publish publicly the encryption key
- You get that key, run it through the algorithm and encrypt the text.
- Send the text to me. I run it through the decryption algorithm and use my private key pair.
- I get the plaintext. Only I can decrypt what you sent me because only I have the private key (right?)
- How do I know that you sent it? Couldn’t someone else just use my public key and send it saying they were you?
- You "sign" what you sent by using your private key and then I use your public key to decrypt the signature to prove it was you that sent it
Diffie-Hellman PK encryption

• Created and patented in 1977
• Whitfield Diffie and Martin Hellman
• Patent expired on April 29, 1997
• Supports exchanging a shared secret key via the algorithm
  • “Based on modular exponentiation using a public base $g$, a public modulus $p$, and private exponents secretly chosen by each participant. For Alice and Bob to agree on a secret key, Alice first chooses a large integer $a$ and Bob chooses a large integer $b$. Alice then calculates $(g^a \mod p) = A$ and sends $A$ to Bob, while Bob calculates $(g^b \mod p) = B$ and sends $B$ to Alice. Alice computes her key as $B^a \mod p$, and it’s identical to Bob’s key, $A^b \mod p$, because both are equivalent to $g^{ab} \mod p$. “
  • [http://www.nyx.net/~awestrop/crypt/dh.htm](http://www.nyx.net/~awestrop/crypt/dh.htm)
Blowfish – private key

- Blowfish is a symmetric block cipher
- Drop-in replacement for DES or IDEA.
- Variable-length key, from 32 bits to 448 bits
- Domestic and exportable use.
- Designed in 1993 by Bruce Schneier as a fast, free alternative to existing encryption algorithms.
- Unpatented and license-free
- [http://www.counterpane.com/bfsverlag.htm](http://www.counterpane.com/bfsverlag.htm)
DES – Digital Encryption Standard

- Adopted by the Federal gov in 1977
- 56-bit key - DES algorithm
- Still widely used by financial services and other industries worldwide
- Cracked July 17, 1998 by EFF DES Cracker, which was built for less than $250,000. It took the machine less than 3 days to complete the challenge.
- Does the government NOT want civilians to have secure encryption?
- [http://www.aci.net/kalliste/des.htm](http://www.aci.net/kalliste/des.htm)
Clipper Chip

• Proposed in April of 1993 by the U.S. government
• A hardware device virtually unbreakable encryption
• The US government would have the key to unlock any information encrypted with the Clipper chip. – escrowed keys
• “If encryption is illegal – only criminals will have encryption”

• http://www-cse.stanford.edu/class/cs201/current/Projects/clipper-chip/intro.html
Digital Hash

- Hash is a “tamper indicator”
  - Message Digest 5 (MD5)
  - Secure Hash Algorithm (SHA)
- Sender takes plaintext message (not encrypted) and runs through an algorithm that gives a hash of the message.
- Sender encrypts the hash with the sender’s secret key (from the PK keys) and you have a digital signature
- Sender encrypts the message text using the Receivers public key (or using a shared secret key)
- The receiver decrypts the message text using her private key
- The receiver decrypts the hash using the senders public key (or the shared secret key). Then runs the hash algorithm on the decrypted text and compares to the original hash to see if the text has been altered.
Digital Signatures

- Use PK encryption but differently.
- I encrypt something with my **private** key and then you use my **public** key to decrypt it. Only I could have encrypted it since only I have the private key.
- How do you know that the public key you got from the internet is indeed the one that I published (and not one put there by a hacker?)
  - Have someone hash my key and digitally sign it (someone like Verisign)
  - Assumes the CA is reputable (a CA is like notary public) they vouch for your identity by signing your documents
Chalkboard discussions

- Hash example - Sum all chars - reversability
- Signing without a hash
- Private vs. public key encryption
- Signature & CAs
  - Verisign & Microsoft Certs problem
- Certificate revocation problem
• Questions
• Comments

• 7.75
Security example

• Writing a check
  – Fill out the draft (the message)
  – Clerk checks your Drivers License & a CreditCard – (two factor Identification and Authentication)
  – Your name is on the check and you have the check book – authorization
  – Write the amount and sign the amount - integrity
  – Sign the check – non-repudiation
  – Clerk takes the check and it goes to the bank – authorization and non-repudiation
Security example – Using a CreditCard

• Present the Card – **Identification**
• Present Driver’s License - **Authentication**
• Clerk runs card through the terminal for authorization for the transaction – talks to your bank (issuing bank via the credit card network) – approves or denies the transaction – if approved may then execute and trigger **settlement** process. Information is transmitted via a dial up connection that may/may not use encryption (possible DES)
• Credit Card may be checked using AVS – address verification service - checks the address you give with the one registered to the card – used with online transaction because the physical card isn’t present. - **authentication**
• Sign the slip – signature – **authorization** and **non-repudiation**
• Credit Card numbers have certain algorithmic checks that they must pass to prevent people just making up numbers or accidentally changing numbers – **authentication / tamper protection**
Processing credit card transactions on the web

Step 1 - The merchant Web server sends an order form to the customer's Web browser.

Step 2 - The customer fills out the form and returns it to the merchant's Web server.

Step 3 - The merchant's Web server sends the credit card number and amount to the issuing bank's computer for verification.

Step 4 - The card issuing bank returns a verification of the customer's credit card purchase.

Step 5 - The merchant's Web server returns the order confirmation to the customer's Web browser.


Copyright Poelman & Associates, Inc. 2002 - All Rights Reserved
SSL

- Secure Sockets Layer
- Created by Netscape
- Combination of Asymmetric and Symmetric encryption
- Uses digital certs for authentication
- Creates a secure – authenticated and encrypted HTTP session over a socket
- HTTPS runs on Port 443 not 80 by default
- Supported in most browsers
- 40 or 128 bit encryption
SSL (2)

- Same basic API as sockets but all data in encrypted and decrypted on the other end.
- SSL Workflow
  - First authenticate via certificate
    - Client
    - Server
    - Or Both
  - Client creates a key for the encrypted session (a secret symmetric key algorithm)
  - Exchanges the secret session key via a PK (public / asymmetric encryption) encryption method using the server’s public key.
  - Session key is only good for a certain amount of time, after it expires a new key is created and repeats the process.
Encryption technologies (optionally) supported by SSL

- **DES**
- **DSA.** Digital Signature Algorithm, part of the digital authentication standard used by the U.S. Government.
- **KEA.** Key Exchange Algorithm, an algorithm used for key exchange by the U.S. Government.
- **MD5.** Message Digest algorithm developed by Rivest.
- **RC2 and RC4.** Rivest encryption ciphers developed for RSA Data Security.
- **RSA.** A public-key algorithm for both encryption and authentication. Developed by Rivest, Shamir, and Adleman.
- **RSA key exchange.** A key-exchange algorithm for SSL based on the RSA algorithm.
- **SHA-1.** Secure Hash Algorithm, a hash function used by the U.S. Government.
- **SKIPJACK.** A classified symmetric-key algorithm implemented in FORTEZZA-compliant hardware used by the U.S. Government. (For more information, see [FORTEZZA Cipher Suites](http://developer.netscape.com/docs/manuals/security/sslin/contents.htm).)
- **Triple-DES.** DES applied three times.

SSL Message Sequence

SSL Messages

Client
1. Client hello
7. Certificate optional
8. Client key exchange
9. Certificate verify optional
10. Change cipher spec
11. Finished
14. Encrypted data

Server
2. Server hello
3. Certificate optional
4. Certificate request optional
5. Server key exchange optional
6. Server hello done
12. Change cipher spec
13. Finished
14. Encrypted data

http://java.sun.com/j2se/1.4/docs/guide/security/jsse/JSSERefGuide.html#SSLOverview © 2002

Copyright Poelman & Associates, Inc. 2002 - All Rights Reserved
Why not use SSL all the time?

• Over head of encryption
  – Use when sending credit card number but not while they are just shopping

• Export restriction limit encryption used between non-US customers and US web sites to 40 bit (not 128). 40 bit is relatively easily cracked.
SSL
(Now called TLS - IETF Standard)

• Can be used with most applications that use TCP
  – Hypertext Transfer Protocol (HTTP)
    • HTTPS / .SHTML
  – Net News Transfer Protocol (NNTP)
  – Telnet
  – Lightweight Directory Access Protocol (LDAP)
  – Interactive Message Access Protocol (IMAP)
  – File Transfer Protocol (FTP)
Kerberos

• Three headed dog that guarded the gates of Hades in Greek Mythology
• A security protocol that provides for mutual authentication between client and server via a Kerberos server.
• Developed by MIT
• Unix OS supported
• Incorporated into Windows XP
• http://web.mit.edu/kerberos/www/
• Questions
• Comments

• 7.5
Java Security APIs
Java Language Security Features

- Strongly typed
- Byte code verification
- Runtime type safety checks
- Class Loaders
- Security Managers
Java Security Approaches

• JVM
  – Byte codes get verified before execution everytime
  – Prevents adding illegal instructions, buffer overflows, pointers to memory
  – No pointers to memory address – your code use opaque references (you can’t get actual me addresses)

• Applets
  – Downloaded to client browser
  – Runs in a “sandbox”
    • No access to resources on the client machine
    • File, Sockets, OS.
  – Only allowed to access sockets on the server that the applet came from
Java Security Approaches

• JSP & Servlet
  – All code executes on the server NOT the client
  – WAR / WEB-INF contains all the code executed, no access to other parts of server file system.
  – WAR can be digitally signed

• Swing
  – Depends on access to box via OS
  – Your code can use the Java Security APIs

• EJBs
  – App Server has security features
  – Roles and Permissions on objects
  – No file access directly
JDK 1.2 Security Model

http://java.sun.com/docs/books/tutorial/security1.2/overview/index.html
Java Roadmap for Security

© Sun 2003
JSSE – Java Secure Socket Extension

• Provides SSL support
• Part of J2SE 1.4v
  – javax.net.ssl.SSLServerSocket
  – javax.net.ssl.SSLSocket
  – Child classes of the standard sockets libraries

• http://java.sun.com/j2se/1.4/docs/guide/security/jsse/JSSERefGuide.html
JCA - Java Cryptography Architecture

• Part of J2SE v1.4
  – DSA, MD5, SHA-1
  – A DSA key pair generator for DSA
  – A DSA key factory
  – Proprietary SHA1PRNG pseudo-random number generation algorithm
  – A certificate path builder and validator for PKIX
  – A certificate store implementation for retrieving certificates and CRLs from Collection and LDAP directories
  – A certificate factory for X.509 certificates and Certificate Revocation Lists (CRLs).
  – A keystore implementation for the proprietary keystore type named JKS.

• Pluggable Cryptographic providers allow updating the algorithms over time.
JCA CSP – Cryptographic Service Provider

- MessageDigest (hash)
- Signature - used to sign data and verify digital signatures.
- KeyPairGenerator - used to generate a pair of public and private keys suitable for a specified algorithm.
- KeyFactory
- KeyStore - keystore is a database of keys
- AlgorithmParameters - manage the parameters for a particular algorithm.
- AlgorithmParameterGenerator
- SecureRandom - generate random or pseudo-random numbers.
- CertPathBuilder - build certificate chains (certification paths).
- CertPathValidator - used to validate certificate chains.
- CertStore - used to retrieve Certificates and CRLs from a repository.
JAAS – Java Authentication and Authorization Service

- Authentication component - reliably and securely determine who is currently executing Java code, regardless of whether the code is running as an application, an applet, a bean, or a servlet.
- Authorization component - providing the means to restrict the executing Java code from performing sensitive tasks, depending on its codesource and depending on who is executing the code.
- Uses Krb5LoginModule (kerberos v5)
- Used to support single signon
GSS API - GENERIC SECURITY SERVICE APPLICATION PROGRAM INTERFACE

• Similar to JSSE but provides Kerberos Support
• Token based versus streams (JSSE), your app must do the communications between parties
• Selective encryption instead of all communications in the session (JSSE)
XML Security

• XML Digital Signature API (JSR 105)
  – Java™ API for signing and validating XML signatures
  – JSR in Public Review

• XML Digital Encryption API (JSR 106)
  – Java™ API for encrypting/decrypting data in XML
  – JSR in Expert Group discussions
• Questions
• Comments

• 8.5
Tracking and Data Mining

- Cash is most anonymous
  - credit, debit, checks aren’t anonymous
- What you bought
- Where you bought it
- How much you spent
- When you bought it can be used to infer things
  - Heath, mental state, illegal income!
    - CIA spy Robert Hansen who sold info to the Soviets was caught by lifestyle that didn’t match his income
  - Purchasing preferences
    - What to try and sell you next – targeted marketing
Interesting

• Federal Driver's Privacy Protection Act (DPPA)
• Federal law that initially authorized the sale of driver's private information.
• Title 18 U.S.C., Chapter 123 Section 2721
• Sponsor, U.S. Representative Barbara Boxer.
• Under the DPPA, states were authorized to sell drivers' names, addresses, birth dates, social security numbers, driver license numbers, digital signatures, and digital photos to private companies for the purpose of developing the centralized registry of identifying information.

• http://www.nydmv.state.ny.us/forms/mv15dppa.pdf
SSN

• 9 digits – 0000000000 to 9999999999
  – Means 1,000,000,000 $1 \times 10^9$ potential numbers
  – Currently $2.54 \times 10^8$ people live in us
  – Many immigrant workers also have them
• People die – what happens to their SSN?
  – Been giving them out since around 1935
  – A generation is approx 20 years so 4 generations – we should be “wrapping around” soon.
• Not guaranteed to be unique
• You can’t get it changed!
  – what if someone steals your identity?
Security has a human and social cost

- Loss of anonymity
- Tracking of data and join of disparately collected data via the unique key (that’s you)
  - Visa (or Master Card or Amex) case
  - Sold two lists.
    - One with names, addresses, phone numbers and CCNum
    - The other with purchases and just the CC Num, no name.
    - “Join” the two lists and you know who spent what
    - Federal government sued them and they were fines $100 MM+!
Summary

• Security
  – Id, Authentication, Authorization, Encryption, Non-repudiation and Access Management

• SSL
  – Secure Sockets Layer
  – HTTPS port 443
  – Mixture of public and private key encryptions
  – Uses digital certs

• Algorithms
  – RSA, SHA, DES, DSA, MD5, Blowfish, Diffie-Hellman, Kerberos

• Java APIs
  – JSSE, JAAS, JCA and GSS API
resources

• Good sun presentation on security
• http://www.verisign.com/
• http://www.thawte.com/
• Interesting page:
  http://archive.ncsa.uiuc.edu/InformationServers/adam/wm96/HACK.HTM
• Enigma simulator written in Java as an applet
  http://homepages.tesco.net/~andycarlson/enigma/enigma_j.html
• Iris scanning http://www.cl.cam.ac.uk/users/jgd1000/roysoc.pdf
• Against the loss of anonymity http://www.networkusa.org/fingerprint.shtml
• http://www.infosecuritymag.com