Security, Privacy, and Cryptography: A Brief Overview

Tadayoshi (Yoshi) Kohno
University of Washington
Goals for this Lecture

• Help you understand why security is important
• Help you understand the common pitfalls in computer security
• Help you understand the mindset and some approaches for overcoming these pitfalls

• Cryptography:
  – Building blocks
  – One-way communications (like PGP)
  – Interactive communications (like SSH)
Why Security?
Views of the Future

Technology has the potential to greatly improve our lives

Technology also has the potential to create new privacy and security risks (and amplify old risks)

Key focus:
  • Anticipate risks with future technologies
  • Address those risks early
(We want to have our cake and eat it too - the promises of new technologies without the risks)
One Example: Personal Medical Devices

- Pacemaker
- Neurostimulator (Urology)
- Neurostimulator (Epilepsy)
- Cochlear Implant
- Drug Pump
- Exoskeleton
One Example: Personal Medical Devices

Trends toward:
- greater computational capabilities;
- longer-range wireless;
- deeper integration into our bodies;
- multi-agent systems.
How Security “Works:”
First Understand Issues with Real Artifacts

With Shane Clark, Benessa Defend, Kevin Fu, Dan Halperin, Tom Heydt-Benjamin, William Maisel, Will Morgan, Ben Ransford
(University of Washington + Harvard Medical School, University of Massachusetts)
Understanding the Issues

- We analyzed an Implantable Cardiac Defibrillator (ICD)
- Related to pacemaker
- Large shock: resync heart
- Monitors heart waveforms

Our model: From 2003
Millions of patients using cardiac devices
Lifecycle

1. Doctor sets patient info
2. Surgically implants
3. Tests defibrillation
4. Ongoing monitoring
5. (Continue use until battery depleted)
Warning

Next part of the talk is targeted at the technical community.

The current risk to patients is small.
Attack #1: Steal Device Programmer
Attack #2: Buy a Device Programmer

On eBay one day last week (10/23/2008):

Intermedics RX2000 ECG Pacemaker Analyzer/Programmer

Buyer or seller of this item? Sign in for your status

5 items found for pacemaker programmer in eBay Stores. Learn more about eBay Stores.

<table>
<thead>
<tr>
<th>Item Title</th>
<th>Price</th>
<th>Shipping to USA</th>
<th>Store</th>
</tr>
</thead>
<tbody>
<tr>
<td>St Jude Model 3510 Pacemaker Programmer / EKG</td>
<td>$99.95</td>
<td>Not specified</td>
<td>The-Printer-Man</td>
</tr>
<tr>
<td>MEDTRONICS 7431 PORTABLE PACEMAKER PROGRAMMER</td>
<td>$400.00</td>
<td>Free</td>
<td>Quality Med Inc</td>
</tr>
<tr>
<td>Medronic 8810 SynchroMed Pacemaker Programmer Complete</td>
<td>$799.00</td>
<td>$32.00</td>
<td>Scott's Attic and More</td>
</tr>
<tr>
<td>Medtronics 9790 Pacemaker Programmer No Reserve!</td>
<td>$1,100.00</td>
<td>Free</td>
<td>Quality Med Inc</td>
</tr>
<tr>
<td>Medronic 9790C Pacemaker Programmer</td>
<td>$1,250.00</td>
<td>Free</td>
<td>Quality Med Inc</td>
</tr>
</tbody>
</table>
Why Steal When You Can Build?

- **Software radio**
- GNU Radio software, $0
- USRP board, $700
- Daughterboards, antennas: $100

~10 cm
(un-optimized)
Attack #3: Eavesdrop Private Info

Implanting physician

Diagnosis

Hospital

Also:
- Device state
- Patient name
- Date of birth
- Make & model
- Serial no.
  ... and more
Attack #4: Sniff Vital Signs

Eavesdropping setup

ICD emits *reconstructible* vital signs

- Issues:
  - Future devices may reveal significantly more information
  - Cryptography does *not* solve the entire problem
Simple Replay Attacks

- Ours: “Deaf” (transmit-only) attacks
- Caveats: Close range; only one ICD model tested; attacks not optimized; takes many seconds
Attack #5: Drain Energy

- Implant designed for **infrequent** radio use
- Radio decreases battery lifetime

“Are you sleeping?”

“No!”
Attack #6: Turn Off Therapies

- “Stop detecting fibrillation.”
- Device programmer would warn here
Attack #7: Affect Patient’s Physiology

• **Induce fibrillation** which implant ignores
• Again, at close range
• In other kinds of implant:
  – Flood patient with drugs
  – Overstimulate nerves, ...
Last part of the talk is targeted at the technical community.

The current risk to patients is small.
Then Develop Defenses
Defenses

• Two parts:
  – Understand context for the system
    • Desired properties for defenses
    • Constraints on defenses
  – Technical mechanisms to build the defenses
• Iterate between these two parts

• Next:
  – Brief survey of both parts
    • “Security Thinking” / “The Security Mindset” / Common pitfalls in security
  – Concrete example: A cryptographic system like SSH
Security and Privacy Crash Course
Computer Security

- **Computer Security** (Informal Definition):
  
  Study of how to design systems that behave as intended in the presence of *malicious third parties*

- Security is different from reliability and safety
  
  - Existence of malicious third party really changes things
  
  - We focus on studying, understanding, anticipating, and defending against these malicious third parties
Security is Non-intuitive

- Our field can be non-intuitive at first:
  - **Mentality**: Bad parties can be skilled, clever, sneaky, and cunning. Not “rational” by most people’s definition. Goal is to cause *intentional* failures.
  - **Imbalance**: Bad parties only need to find *one* way to compromise the security of your system; defender must defend against *all* realistic attack vectors
  - **Unpredictability**: Bad parties “*win*” by doing what the defenders don’t expect. Common expression: “Anyone can design a system that they themselves cannot break.”

- Next few slides: Survey common themes in security
Threat Modeling

- Security is about *threat modeling*:
  - Who are the potential attackers?
  - What are their resources and capabilities?
  - What are their motives?
  - What assets are you trying to protect?
  - What might the attackers try to do to compromise those assets?

- Need to answer these questions early, before you can even begin to make any conclusions about a real system.
Common Fallacy #1

- **Common fallacy #1**: “A system is either secure or insecure.”

- Security is a gradient
- No such thing as a “perfectly secure system”
  - All systems are vulnerable to attacks
  - We’re interested in the *level* of security that a system provides (recall threat model)

- **Our suggestion**: Need for industry-wide definition of what it means for an IMD to provide a sufficient level of security
Common Fallacy #2

- **Common Fallacy #2**: “There’s never been an attack in the past, so security is not an issue”
  - Many variants, like: “There’s never been an attack in the past, so there won’t be in the future”

- Above reasoning is *intuitive* but also *incorrect*.
- Equivalent to
  - “I’ve never been robbed, so I don’t need to lock my front door.”
- Problems with this:
  - It might have happened, you just don’t know because you haven’t been worrying about it.
  - Technology changes capabilities, incentives, and context so always new things attackers might do
Common Fallacy #2

- Example: Ping-of-Death
  - When Microsoft created Windows 95, the developers thought that something “would never happen”
  - But then the Internet evolved, Windows 95 machines were hooked to the Internet ... and ... it happened!
  - Result: What’s called the Ping-of-Death
Common Fallacy #3

- **Common Fallacy #3**: “We use proprietary security algorithms, so the bad guys won’t know these algorithms and our system is secure.”

- **Flaw #1**: Bad guys can learn these algorithms
  - Insiders, consultants, dumpster divers, corporate espionage, terrorists, ...
  - Bad guys could reverse engineer algorithms

- **Flaw #2**: Security through obscurity
  - Proprietary algorithms have a history of being less secure than standardized algorithms
  - Recall saying “anyone can design a system they themselves cannot break”
  - If it’s proprietary, how can outsiders (public, FDA, etc) know for sure?
MiFare RFID crack more extensive than previously thought

Seconds, not hours, to effect; plus version tappable too

By Geeta Dayal

April 15, 2008 (Col)

— used daily by mi passes and other; thought, according development Tues conference in Ista

Mere seconds are few hours, as estir graduate student a reverse-engineerin takes only 12 sec an ordinary laptop.
Common Fallacy #4

Common Fallacy #4: “We’re secure because we use standardized security algorithms like RSA, AES, SSL, ...”

Using standardized algorithms is a good, but far from sufficient

Analogy:
- Standardized security algorithms are like standardized locks
- Locks themselves may be strong, but security of building depends on many other things (how you key the locks, how you attach locks to door, how door frame is mounted, whether you also lock the windows, etc)

Many examples, e.g.,
- Diebold Voting Machines
Common Fallacy #5

- **Common Fallacy #5**: “We’ve addressed all known security concerns, so our system is now secure”

- From my own work:
  - 2003: We identified security problems with the Diebold voting machine
  - 2004: Diebold introduced defenses to that specific attack; RABA re-evaluated and found that the fix introduced a new security vulnerability
  - 2007: Diebold introduced defenses to that new attack; we re-evaluated and found that the second fix introduced another new security vulnerability
Common Fallacy #6

- **Common Fallacy #6**: “Our system is secure because we’ve had it analyzed by third-party testing authorities”

- History in other fields says otherwise; consider e-voting:
  - E-voting machines are regularly evaluated by third-party testers
  - But researchers are regularly finding security flaws with these systems
Common Fallacy #7

- **Common Fallacy #7**: “Our system must provide an ‘acceptable level of security’ since we’ve had it analyzed by one or more security experts or teams”
- Definitely a good sign, but not sufficient
- Different security firms have different levels of expertise; security firms also often lack medical domain knowledge
- **Who defines what an “acceptable level of security is”**
  - Does the vendor? Do the security consultants?
  - Each of the above parties have limited vantage points
  - **Our belief**: Only the FDA is in a position to have a global view at regulating what constitutes an acceptable level of security
Security Problems with Security Software

• History is full of products from *security companies* that have security vulnerabilities

• Conclusions:
  – Security is hard, even for security experts
  – Need for industry-wide oversight
  – Also need many people focusing on this problem
Known Vulnerabilities in Firewalls

[Replication or Save Conflict]

<table>
<thead>
<tr>
<th>ID</th>
<th>Date Public</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>VU#508209</td>
<td>09/07/2005</td>
<td>Check Point Firewall rules may improperly handle network traffic</td>
</tr>
<tr>
<td>VU#639507</td>
<td>10/01/2001</td>
<td>Cisco PIX Firewall Manager stores enable password in plain text</td>
</tr>
<tr>
<td>VU#310295</td>
<td>07/09/2001</td>
<td>Check Point RDP Bypass Vulnerability</td>
</tr>
<tr>
<td>VU#45716</td>
<td>04/28/2003</td>
<td>Kerio Personal Firewall vulnerable to buffer overflow</td>
</tr>
<tr>
<td>VU#258731</td>
<td>10/08/2001</td>
<td>Check Point VPN-1/FireWall-1 4.1 on Nokia IPXXX firewall appliance retransmits original packets</td>
</tr>
<tr>
<td>VU#210927</td>
<td>03/19/2003</td>
<td>IBM Tivoli Firewall Toolbox contains vulnerability</td>
</tr>
<tr>
<td>VU#20825</td>
<td>07/11/2000</td>
<td>Cisco Secure PIX Firewall TCP Reset Vulnerability</td>
</tr>
<tr>
<td>VU#441078</td>
<td>09/22/2004</td>
<td>Symantec Firewall/VPN appliance vulnerable to DoS via UDP port scan</td>
</tr>
<tr>
<td>VU#35958</td>
<td>06/05/2000</td>
<td>IP Fragmentation Denial-of-Service Vulnerability in FireWall-1</td>
</tr>
<tr>
<td>VU#5053</td>
<td>08/31/98</td>
<td>Older Versions of Cisco PIX Firewall Manager permits retrieval of files</td>
</tr>
<tr>
<td>VU#236045</td>
<td>09/07/2005</td>
<td>Cisco IOS Firewall Authentication Proxy vulnerable to buffer overflow via specially crafted user authentication credentials</td>
</tr>
<tr>
<td>VU#362483</td>
<td>11/28/2001</td>
<td>Cisco IOS Firewall Feature Set fails to check IP protocol type thereby allowing packets to bypass dynamic access control lists</td>
</tr>
<tr>
<td>VU#641012</td>
<td>04/28/2003</td>
<td>Kerio Personal Firewall vulnerable to replay attack</td>
</tr>
<tr>
<td>VU#682110</td>
<td>05/12/2004</td>
<td>Multiple Symantec firewall products fail to properly process DNS response packets</td>
</tr>
<tr>
<td>VU#539363</td>
<td>10/15/2002</td>
<td>State-based firewalls fail to effectively manage session table resource exhaustion</td>
</tr>
<tr>
<td>VU#634144</td>
<td>05/12/2004</td>
<td>Multiple Symantec firewall products fail to properly process NBNS response packets</td>
</tr>
<tr>
<td>VU#6733</td>
<td>07/15/98</td>
<td>PIX 'established' and 'conduit' command may have unexpected interactions</td>
</tr>
<tr>
<td>VU#637318</td>
<td>05/12/2004</td>
<td>Multiple Symantec firewall products contain a buffer overflow in the processing of DNS resource records</td>
</tr>
<tr>
<td>VU#294998</td>
<td>05/12/2004</td>
<td>Multiple Symantec firewall products contain a heap corruption vulnerability in the handling of NBNS response packets</td>
</tr>
<tr>
<td>VU#435358</td>
<td>07/28/2004</td>
<td>Check Point VPN-1 products contain boundary error in the ASN.1 decoding library</td>
</tr>
<tr>
<td>VU#446680</td>
<td>12/19/2000</td>
<td>Check Point FireWall-1 allows fragmented packets through firewall if Fast Mode is enabled</td>
</tr>
<tr>
<td>VU#749870</td>
<td>08/03/2004</td>
<td>Juniper Networks NetScreen firewall contains a DoS vulnerability in the SSHv1 service</td>
</tr>
</tbody>
</table>
Common Fallacy #8

- **Common Fallacy #8**: “If we increase security, we’d be forced to decrease safety and/or usability”
- Challenging, but not impossible
- To make educated decisions and arguments we need to:
  - explore solution space,
  - gauge what’s possible, and
  - assess levels of security and usability provided by different solutions
Common Fallacy #9

- **Common Fallacy #9**: “We don’t need end-devices (like IMDs) to be secure because the back-end system is already secure”
- Expression in security community: “Security only as strong as the weakest link”
- We need to consider security of *all* aspects of the overall system
Common Fallacy #10

- **Common Fallacy #10**: “Only sophisticated adversaries will be able to successfully attack our system”
- Expression in security community:
  - “Attacks only get better, easier to mount over time”
- Some adversaries will be sophisticated (we return to this later)
- Different actors: Sophisticated bad guys create tools that less sophisticated bad guys use
Common Fallacy #11

- **Common Fallacy #11**: “*Insiders are not going to be adversaries*”
- Plenty of examples to the contrary (although companies don’t like to talk about it)

- Spies
- Greedy employees
- Disgruntled ex-employees
- ...
A former Boeing engineer was arrested on Monday on charges of stealing trade secrets for China related to several US aerospace programmes, including the space shuttle, the US Justice Department said.

It also announced a separate case in which a US Defense Department official and two others were arrested on Monday on espionage charges involving the passing of classified US government documents to China.

Previous spy cases involving China and the US include:

• 1999 – Los Alamos National Laboratory, where the first US nuclear bombs were developed in the 1940s, comes under fire over security after US prosecutors charge scientist Wen Ho Lee with 59 counts of illegally downloading nuclear weapons data onto portable tapes and...
Teller nabbed in counterfeit bill scheme

By Scott Merzbach
Staff Writer
Published on February 03, 2006

It sounded like the oldest trick in the book.

Two bank tellers accused of stealing customers' money

Associated Press
Saturday, February 23, 2005

Two Steamboat Springs bank tellers are accused of stealing about $1.2 million from customers' accounts.

Pamela Jean Williams and Terri Dawn Moody Fatka were arrested Thursday and released on $20,000 bail.

They were arrested on suspicion of felony theft and forgery.

Each faces up to 15 years in prison if convicted.

Police Capt. Joel Rae said that five victims have been identified but that more may come forward. Police urged customers to check their bank statements.
Common Fallacy #12

• Common Fallacy #12: “We’ve thought of everything”
• Doesn’t apply to computer security - can never *prove* to yourself that you’ve thought of all attackers
• Same thing applies to these slides: This list of common fallacies is not exclusive
Potential Security Goals

- Availability
- Integrity
  - Data
  - Settings and software
- Privacy
  - Device existence
  - Device type
  - Specific-device ID privacy
  - Measurement and log privacy
  - Bearer privacy
Attacker Resources

• Insiders
• Outsiders
• Coordinated Attackers
• Commercial Equipment
• Custom Equipment
Potential Motives

- Why would someone want to compromise the security of an IMD?
- Example motives:
  - Terrorism (lots of anger toward US citizens)
  - Random acts of violence
  - Foreign government or military action
  - Malice towards company (e.g., ex-employee, competitor or new startup)
  - Malice towards individuals
  - Surveillance
  - Identity theft and stealing private information
  - Self-prescription (“body hacking”, morphine dosage)
Cyber Terrorism and Foreign Nations

- Terrorism is a real concern - both at home and abroad
  - Attacking medical devices is a potential form of cyber terrorism

- Even threat of an attack - even if never mounted - could cause serious harm

- Cyber-armies in foreign nations:
  - Well funded, incredibly smart and technically skilled
China’s cyber army is preparing to march on America, says Pentagon

Chinese military hackers have prepared a detailed plan to disable America’s aircraft battle carrier fleet with a devastating cyber attack, according to a Pentagon report obtained by The Times.
Random and Malicious Acts

• Unfortunately, people do mount random and malicious acts of violence.
Polish teen derails tram after hacking train network

By John Leyden
Published Friday 11th January 2008 11:56 GMT

A Polish teenager allegedly turned the tram system in the city of Lodz into his own personal train set, triggering chaos and derailing four vehicles in the process.

April 30, 2008

Heparin Contamination May Have Been Deliberate, F.D.A. Says

By GARDINER HARRIS

WASHINGTON — Federal drug regulators believe that a contaminant detected in a crucial blood thinner that has caused 81 deaths was added deliberately, something the Food and Drug Administration has only hinted at previously.
"I was able to trace back the source of the attack to a handful of sites where the perpetrators were instigating the event," said Bernard Ertl, CWE Administrator. "It was just a bunch of very immature people delighting in their attempts to cause people misery. Attacking sites is just a way to pass time for them. Unfortunately, this time they tried to hurt people. Seizures are not a laughing matter. A member of CWE passed away just two weeks ago from a seizure. SUDEP (Sudden Unexplained Death in Epilepsy) is a very real and serious concern."
“This was clearly an act of vandalism with the intent to harm people, and we shut the attack down immediately,” said Eric R. Hargis, president and CEO of the Epilepsy Foundation.
Implications to IMDs

- Observation:
  - Epilepsy patients remotely attackable
    - Their “attack surface” is large
  - People *have* exploited this fact to try to hurt them
  - “Attack surface” for other patients may increase as IMDs become more sophisticated and communicative

- Conclusion:
  - We need to be *carefully consider* future similar acts to IMD patients
802.11 WiFi Sniper Yagi
Uninvited Radio Suitcases

Attacking Own Device: Body Hacking

Magnet implanted under finger to give person “sixth sense” (Quinn Norton)

Warning: Be careful if you google “body hacking”
Cryptography:
Let’s look at SSH’ and PGP’

SSH’ and PGP’ are “like” SSH and PGP
Common Communication Security Goals

**Privacy** of data
Prevent exposure of information

**Integrity** of data
Prevent modification of information
Symmetric Setting

Both communicating parties have access to a shared random string $K$, called the key.
Asymmetric Setting

Each party creates a public key $pk$ and a secret key $sk$. 

Alice: $pk_A, sk_A$

Bob: $pk_B, sk_B$

Adversary: $pk_B, sk_B$
Achieving Privacy (Symmetric)

Encryption schemes: A tool for protecting privacy.

Diagram:
- Alice (M) -> Encrypt (K) -> Ciphertext (C) -> Decrypt (K) -> Bob (M)
- Message (M) -> Ciphertext (C)
- Adversary

Symbols:
- M: Message
- K: Key
- C: Ciphertext
Achieving Privacy (Asymmetric)

Encryption schemes: A tool for protecting privacy.

Alice
\(\text{pk}_A, \text{sk}_A\)

Bob
\(\text{pk}_B, \text{sk}_B\)

Message \(\ldots M\)

Ciphertext \(\ldots C\)

Adversary
Achieving Integrity (Symmetric)

Message authentication schemes: A tool for protecting integrity.
(Also called message authentication codes or MACs.)
Achieving Integrity (Asymmetric)

Digital signature schemes: A tool for protecting integrity and authenticity.
Getting keys: PBKDF

Password-based Key Derivation Functions

Alice

Password → PBKDF → K (Key check value)
Getting keys: Key exchange

**Key exchange protocols:** A tool for establishing a share symmetric key
Getting keys: CAs

Each party creates a public key $pk$ and a secret key $sk$.

(Public keys signed by a trusted third party: a certificate authority.)

Alice

$pk_A, sk_A$

$pk_B, \text{sign}(sk_{CA}, B, pk_B)$

Bob

$pk_B, sk_B$

$pk_A, \text{sign}(sk_{CA}, A, pk_A)$

Adversary
“Random” Numbers

Pseudorandom Number Generators (PRNGs)

Alice

PRNG

R₁, R₂, R₃, R₄, R₅, ...

Machine State

User Input

... 

Adversary
In the rush to clean up the Debian OpenSSL fiasco, a number of other major security holes have been uncovered:

<table>
<thead>
<tr>
<th>Affected System</th>
<th>Security Problem</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fedora Core</td>
<td>Vulnerable to certain decoder rings</td>
</tr>
<tr>
<td>Xandros (EEE PC)</td>
<td>Gives root access if asked in stern voice</td>
</tr>
<tr>
<td>Gentoo</td>
<td>Vulnerable to flattery</td>
</tr>
<tr>
<td>OLPC OS</td>
<td>Vulnerable to Jeff Goldblum's powerbook</td>
</tr>
<tr>
<td>Slackware</td>
<td>Gives root access if user says Elvish word for &quot;friend&quot;</td>
</tr>
<tr>
<td>Ubuntu</td>
<td>Turns out distro is actually just Windows Vista with a few custom themes</td>
</tr>
</tbody>
</table>
One-way Communications

PGP is a good example

Message encrypted under Bob’s public key
Interactive Communications

In many cases, it’s probably a good idea to just use a standard protocol/system like SSH, SSL/TLS, etc...

Let’s talk securely; here are the algorithms I understand

I choose these algorithms; start key exchange

Continue key exchange

Communicate using exchanged key
Let’s Dive a Bit Deeper
One-way Communications

(Ininformal example; ignineg, e.g., signatures)
1. Alice gets Bob’s public key; Alice verifies Bob’s public key (e.g., via CA)
2. Alice generates random symmetric keys K1 and K2
3. Alice encrypts the message M the key K1; call result C
4. Alice authenticates (MACs) C with key K2; call the result T
5. Alice encrypts K1 and K2 with Bob’s public key; call the result D

6. Send D, C, T

(Assume Bob’s private key is encrypted on Bob’s disk.)

7. Bob takes his password to derive key K3
8. Bob decrypts his private key with key K3
9. Bob uses private key to decrypt K1 and K2
10. Bob uses K2 to verify MAC tag T
11. Bob uses K1 to decrypt C
One-way Communications

*(Informal example; e.g., signatures)*

1. Alice gets Bob’s public key (e.g., via CA)
2. Alice generates random symmetric keys K1 and K2
3. Alice encrypts the message M with the key K1; call result C
4. Alice authenticates (MACs) C with key K2; call the result T
5. Alice encrypts K1 and K2 with Bob’s public key; call the result D

(As assumed, Bob’s private key is encrypted on Bob’s disk.)

7. Bob takes his password to derive key K3
8. Bob decrypts his private key with key K3
9. Bob uses private key to decrypt K1 and K2
10. Bob uses K2 to verify MAC tag T
11. Bob uses K1 to decrypt C

*Be Careful About Trying This On Your Own*

*(Details Omitted; Easy to Get Wrong; ...)*
Interactive Communications

*Informal example; details omitted*

1. Alice and Bob exchange public keys and certificates
2. Alice and Bob use CA’s public keys to verify certificates and each other’s public keys
3. Alice and Bob take their passwords and derive symmetric keys
4. Alice and Bob use those symmetric keys to decrypt and recover their asymmetric private keys.
5. Alice and Bob use their asymmetric private keys and a key exchange algorithm to derive a shared symmetric key
   (They key exchange process will require Alice and Bob to generate new pseudorandom numbers)
6. Alice and Bob use shared symmetric key to encrypt and authenticate messages
   (Last step will probably also use random numbers; will need to rekey regularly; may need to avoid replay attacks,...
Interactive Communications

(*Informal example; details omitted*)

1. Alice and Bob exchange public keys and certificates
2. Alice and Bob use CA's public keys to verify certificates and each other’s public keys
3. Alice and Bob take their passwords and derive symmetric keys
4. Alice and Bob use those symmetric keys to decrypt and recover their asymmetric private keys
5. Alice and Bob use their asymmetric private keys and a key exchange algorithm to derive a shared symmetric key
   (They key exchange process will require Alice and Bob to generate new pseudorandom numbers)
6. Alice and Bob use the shared symmetric key to encrypt and authenticate messages

*Be Careful About Trying This On Your Own*  
(Details Omitted; Easy to Get Wrong; ...)*
Some Attacks to Consider

• Chosen-plaintext attacks
• Chosen-ciphertext attacks

• Replay attacks
• Reordering attacks

• Protocol-rollback attacks