Distributed Systems

Rik Sarkar
James Cheney

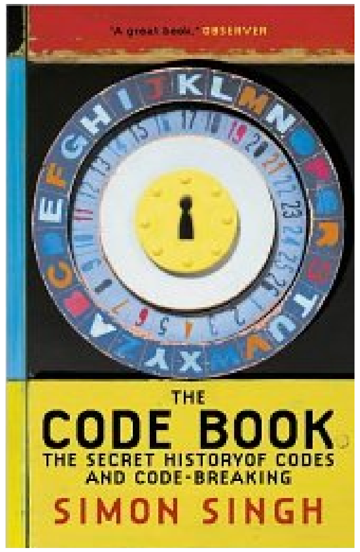
Security & Cryptography
March 13, 2014

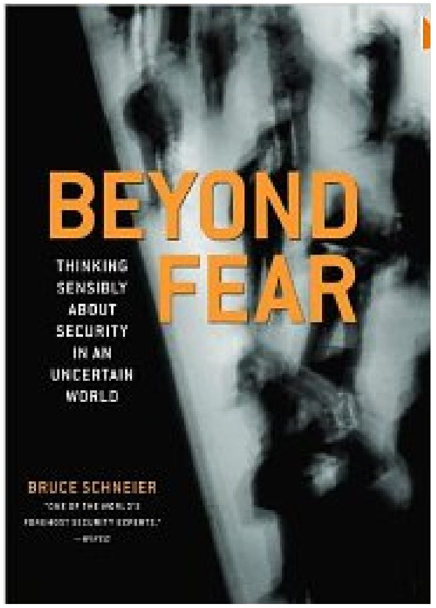
Overview

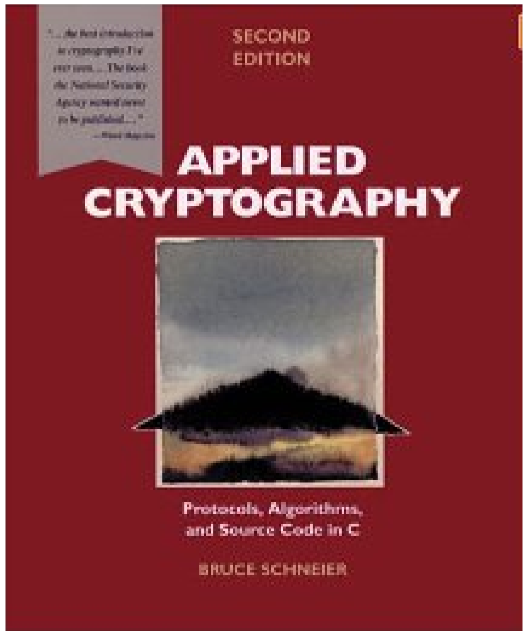
- In this part of the course we will look at **security** in distributed systems
- **Cryptography** will provide the basis of secrecy and integrity
  - That is, making sure that no unauthorised entity may read any particular message
  - No unintended message is delivered, including a duplication of an intended message
- We will examine private-key techniques as well as public-key techniques and digital signatures
- Will also cover other security issues arising in distributed systems

Books

- We will focus on threats to distributed systems caused by the unavoidable exposure of their communication channels
- The largest threat is generally human error
- Bruch Schneier also has a newsletter each month called "cryptogram" which talks about many security related topics including cryptography and physical/human related policies







Cryptography

- Computer security and computer cryptography are separate subjects
  - cryptography provides the basis for most of the mechanisms that we use in computer security
- Until the 1990s cryptographic techniques were tighylt controlled military/government
- When Bruce Schneier first published his book “Applied Cryptography” in 1994 the legal status of including cryptographic algorithms and techniques was in doubt.
- Today it is recognized that "security through obscurity" doesn't work; algorithms are peer-reviewed and publicly available

Pre-1999 US Munitions Control

- RSA crypto-algorithms, were, until 1999, classified by the US State Department as **munitions**
  - Meaning they were classified in the same category as: chemical and biological weapons, tanks, heavy artillery, and military aircraft
- Additionally this meant that it was illegal to export such cryptographic algorithms, with penalties including $1m fines and long prison sentences
- This was obviously silly:
  - It is impossible to enforce
  - The technology is widely available throughout the world
  - Algorithms published in international journals
  - Some cryptographic algorithms were developed outside the US

Pre-1999 US Munitions Control

- Popular email programs such as Netscape Communicator had to have separate downloads for US based downloaders and external downloaders
- When it went open-source and became Mozilla the external versions were quickly patched to include full 160-bit encryption
- People took to methods of highlighting how ridiculous export ban was
- one such effort demonstrated that RSA crypto algorithms can be written in a fairly short amount of Perl code:

```
#!/bin/perl -sp0777i<X+d*lMLa^*lN%0]dsXx++lMlN/dsM0<j]dsj
$/=unpack('H*',$_);$_=`echo 16dio\U$k"SK$/SM$n\EsN0p[lN*1
lK[d2%Sa2/d0$^Ixp"|dc`;s/\W//g;$_=pack('H*',/((..)*)$/)
```

Pre-1999 US Munitions Control

- So to highlight how ludicrous it was people started attaching it to emails
- Technically if said emails were sent outside the US such people could have been prosecuted

```
--
The following is classified as munitions by
 the US state department:
#!/bin/perl -sp0777i<X+d*lMLa^*lN%0]dsXx++lMlN/dsM0<j]dsj
$/=unpack('H*',$_);$_=`echo 16dio\U$k"SK$/SM$n\EsN0p[lN*1
lK[d2%Sa2/d0$^Ixp"|dc`;s/\W//g;$_=pack('H*',/((..)*)$/)
```

We will assume

- Wherever you are in the world you have access to cryptographic protocols and algorithms
- There are a set of nodes which share resources
- Resources may be physical or data/programming objects
- Communication is via message passing only, and hence access to shared resources occurs via message passing
- The nodes are connected via a network which may be accessed by any **attacker**
- An attacker may copy or read any message transmitted through the network
- They may also inject arbitrary messages, to any destination purporting to come from any source

Policies and Mechanisms

- There is an important distinction between a security **policy** and a security **mechanism**
- Policy: what you want to achieve
- Mechanism: what you actually do
- For example, the door to your accommodation is likely secured using a lock and key, that is the security mechanism
  - Additional rules such as "lock door when you leave" and "change locks if key lost" may be needed too
- The policy is "prevent people from entering accommodation"
- Another mechanism for the same policy: door is broken, but security guard keeps people from entering

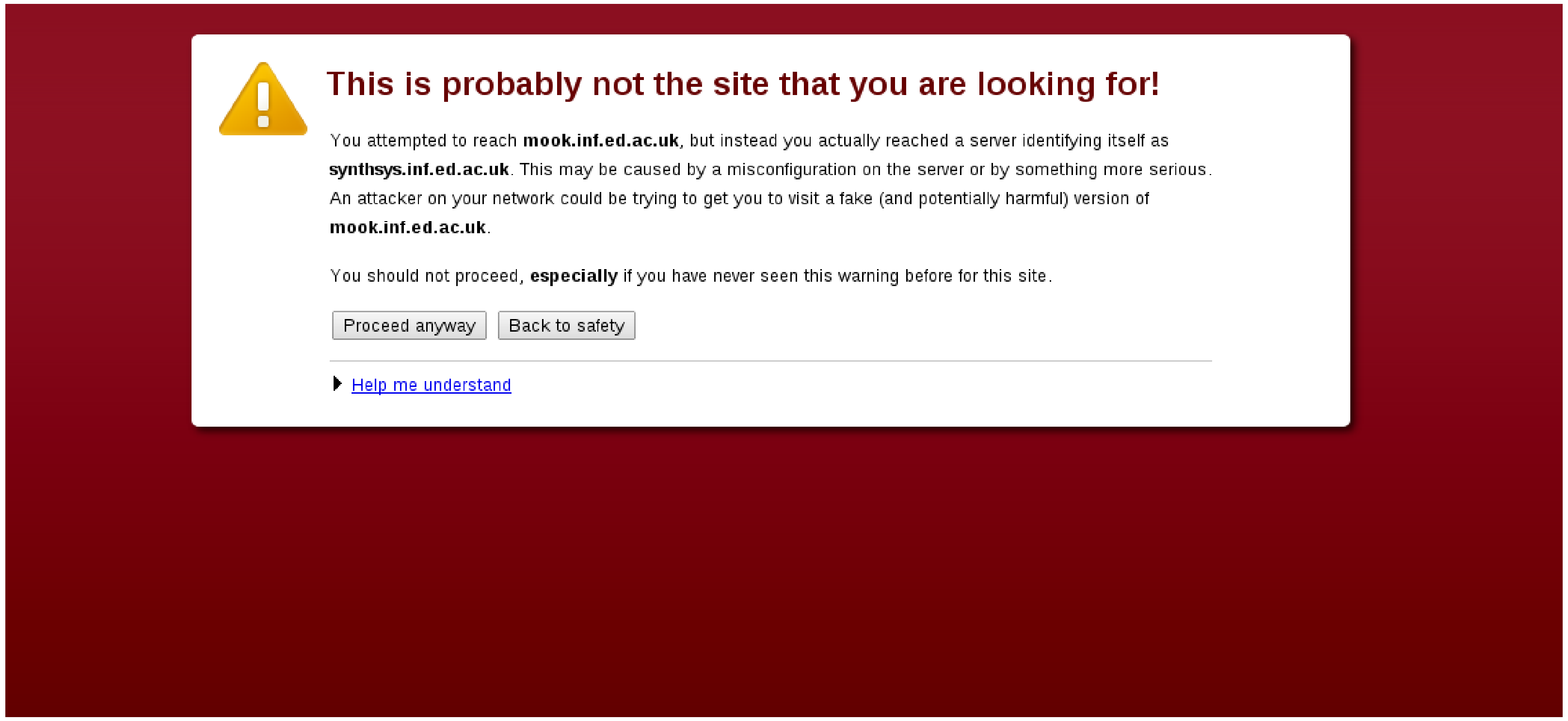
Threats and Attacks

- For most types of network, an attacker wishing to obtain private information can simply listen in on all messages
  - especially wireless
- Doing so means that it is relatively simple to log all messages between communicating computers
- Depending on the application simply knowing the contents of some messages may be enough,
- otherwise the attacker may need information about the distributed algorithm in question in order to construct information from the data in the messages that were recorded

Threats and Attacks

- A slightly more elaborate attack is to construct a server in between the client and the intended server
- If the client does not authenticate the server, then it may send private information to what it believes to be the intended server
- Often the fake server will then log the information sent to it, but then also forward it on the real server in question
  - "man in the middle"
- Thus the attack is non-trivial to detect.
- This is a common technique for obtaining web-passwords

Security



- Third party “Certificate Authorities” issue digital certificates containing encryption keys to verify the identity of secure websites

Threats and Attacks

- Threats and attacks fall into three broad categories:

1. Leakage
   - The acquisition of data by unauthorized entities
2. Tampering
   - The alteration of data by an unauthorized entity
3. Vandalism
   - Disruption to the service in question without gain to the perpetrators

Threats and Attacks

- We can further distinguish attacks in a distributed system by the way in which communication channels are misused:

1. Eavesdropping
   - Obtaining copies of messages without authority
2. Masquerading
   - Sending or receiving messages using the identity of another process/entity without their authority
3. Message Tampering
   - Intercepting messages and altering them before forwarding them on to their intended recipient
4. Replaying
   - Storing intercepted messages and sending them at a later date. This attack can be effective even when used against authenticated and encrypted messages.
5. Denial of Service
   - Flooding a service with requests such that it cannot handle legitimate requests

Information Existence

- Even with strong encryption, the detection of a message transmitted between two processes may leak information
- The mere existence of such a message may be the source of information.
  - "Alice said something to Bob, what are they up to?"
  - A flood of messages to a dealer of a particular set of stocks may indicate a high-level of trading for a particular stock
- One possible defense is to regularly send random nonsense/ignorable messages

Trade-offs

- Ultimately all security measures involve trade-offs
- A cost is incurred in terms of computational work and network usage for use of cryptography and other protocols
  - How many passwords do *you* have?
- Where a security measure is not correctly specified it may limit the availability of the service for legitimate users/uses
- These costs must be compared against the threat or cost of failure to maintain security
- Generally we wish to avoid disaster and minimize mishaps
  - while still allowing effective use of the system

Assume the worst

- Interfaces are exposed distributed systems are designed such that processes offer a set of services, or an interface.
- These interfaces must be open to allow for new clients. Attackers therefore are able to send an arbitrary message to any interface
- Networks are insecure: An attacker can send a message and falsify the origin address so as to masquerade as another user.
- Host addresses may be spoofed so that an attacker may receive a message intended for another
- Algorithms and program code is available to attackers
- Messages sent may be intercepted but that may not be useful since to make sense of the message an attacker may need to know the purpose/protocol within which the message is sent.
  - Assume that attacker does know these things and has significant computational resources - no "security through obscurity"

Assume the worst

- Attackers may have access to large resources
  - Do not therefore rely on the fact that you may compute something faster than an attacker,
  - or that an attacker has a limited timeframe in which their attack may be valid/dangerous/worthwhile
- Assume all code may have flaws
  - Minimize the part of your software responsible for security must be trusted.
  - Often called the trusted computing base (TCB)
- Principle of least privilege
  - don't give unnecessary abilities to participants that don't need them, to prevent accidents/misuse
  - "Administrator" vs. normal user

Cryptography

- Modern Cryptography relies on the use of algorithms which distort a message and reverse that distortion using a secrets called keys
- A simple substitution cipher like ROT13 is an example of this:
- In this case the key is the mapping of characters: a→n,b→o,c→p,...
- Today’s encryption techniques are believed to have the property that the decryption key cannot be feasibly guessed using the ciphertext (the encrypted message)

Cryptography

- There are two main types of algorithms in use:
- 1. shared secret keys
  - both parties must share knowledge of the secret key and it must not be shared with any other party
- 2. public/private key pairs
  - The sender uses the receiver’s public key to encrypt the message.
  - The encryption cannot be reversed by the public key and can only be reversed by the receiver’s private key
  - The sender needs to know the receiver’s public key but need not know the receiver’s private key
  - Anyone may know the receiver’s public key but the private key must be known only to the receiver
- Both kinds of algorithms are very useful and widely used
  - public/private key algorithms require 100/1000 times more processing power
  - The need for initial secure transfer of the private key often outweighs the disadvantage

Some Notation and Characters

- Alice and Bob are participants in security protocols
- Carol and Dave are extra participants for 3,4 party protocols
- Eve is an eavesdropper
- Mallory is a malicious attacker
- Sara is a server

- Alice has the secret key K_A and Bob the secret key K_B
- They have a shared secret key K_{AB}
- Alice has a private key K_{Apriv} and a public key K_{Apub}
- $\{M\}_K$ is a message encrypted with key K
- $[M]_K$ is a message signed with key K

Scenario 1. Secure communication

- Cryptography can be used to enable secure communication
- In this scenario each message is encrypted and can only be decrypted with the correct secret key
- So long as that secret key is not compromised then secrecy can be maintained
- Integrity is generally maintained using some redundant information within the encrypted message, such as a checksum
  - Ensure that encrypted message hasn't been corrupted.
  - Often obvious from message being garbage (but may be hard to detect in general).

Scenario 1. Secure communication

- Alice wishes to send some secret information to Bob
- If they share the secret key K_{AB} then:
  - Alice uses the key and an agreed encryption algorithm $E(K_{AB},M)$ to encrypt and send any number of messages $\{M_i\}K_{AB}$
  - Bob decrypts the messages using the corresponding decryption algorithm $D(K_{AB},M)$
- Two problems:
  - How can Alice initiate this communication by sending the secret key K_{AB} to Bob securely?
  - How does Bob know that a message $\{M_i\}$ isn't a copy of an earlier encrypted message sent by Alice but intercepted by Mallory?

Scenario 2. Authentication

- Cryptography can be used to authenticate communication between a pair of participants
- If there is a shared secret key known only to two parties, then a successful decryption of a received message requires that the message was originally encrypted using the appropriate key
- If only one (other) party knows of that secret key then we can deduce from whom the message originated

Scenario 2. Authentication

- Alice wishes to communicate with Bob
- Sara is a securely managed authentication server
- Sara stores a secret key for each user, each user knows (or can generate from a password) their own secret key.
- Sara may generate a ticket which consists of a new shared key together with the identity of the participant to whom the ticket is issued

Steps to secure communication:

- Alice sends a request to Sara stating who she is and requesting a ticket for secure communication with Bob.
- Sara creates a new secret key K_{AB} to be shared between Alice and Bob.
- Sara encrypts the ticket using Bob's secret key and sends that together with the secret key all encrypted with Alice's secret key $\{(\{ticket\}K_B , K_{AB})\}K_A$
- Alice decrypts this message and obtains the shared secret key and a message containing the ticket encrypted using Bob's secret key.
  - Alice cannot decrypt this ticket message
- Alice sends the ticket together with her identity and a request for shared communication to Bob
- Bob decrypts the ticket: $\{(K_{AB} , Alice)\}K_B$, confirms that the ticket was issued to the sender (Alice).
- Alice and Bob can then communicate securely using the (now) shared secret key K_{AB}. Generally the key is used for a limited amount of time before a new one is requested from Sara.

Scenario 2. Authentication

- This is a simplified version of Needham and Schroeder algorithm which is used in Kerberos system (developed at MIT and used here)
- The simplified version does not protect against a replay attack, where old authentication messages are replayed
- It is used within organizations since the individual private keys, K_A, K_B etc, must be shared between the authentication server and the participants in some secure way
- It is therefore inappropriate for use with wide area applications such as eCommerce
- An important breakthrough was the realization that the user's password need not be sent through the network each time authentication is required.
  - Instead "challenges" are used
  - When the server sends Alice the ticket and new shared private key it encrypts it with K_A, which is based on Alice's password
  - An attacker pretending to be Alice would be defeated at this point