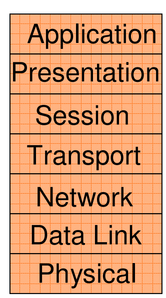
CSE/EE 461 – Module 15

Security

This Time

- Network security
- Focus
  - How do we secure distributed systems?
- Topics
  - Privacy, integrity, authenticity, timeliness
  - Cryptography
  - Practical security



Preliminaries: End-Host Security

- Traditional security concepts:
  - Integrity
    - My files shouldn't be modifiable by an unauthorized user
  - Privacy
    - My files shouldn't be readable by an unauthorized user
- Traditional security mechanisms:
  - Authentication
    - Who are you?
  - Authorization
    - What are you allowed to do?

Preliminaries (cont.)

- "Trusted computing base"
  - Components of the system that you believe are respecting the security policy but that are not verified as doing so
    - The user trusts the operating system
      - E.g., won't leak your files to unauthorized users, won't spuriously delete/modify them
- User trusts applications
  - Emacs isn't mailing your file to its authors
- User trusts the hardware
  - Is your keyboard trustworthy?
  - Is an ATM trustworthy?
- Does the OS trust users?
  - Mandatory access control

Preliminaries: Network Security

- Most of the technologies in lower protocol layers were developed pre-Internet
- Pre-Internet:
  - There weren't many network services (telnet, mail, ftp, a few others)
  - There weren't many machines on networks
    - Many local networks, but not very interconnected
  - "End-to-end security" made sense
    - Trusted OSes running trusted applications run by trusted users
      - At the very least, you could probably track down a malicious user
- Result: no security mechanisms were built into protocols themselves
  - E.g., mail spoofing was trivial

Preliminaries: Post-Internet

- Really an entirely new situation
  - Servers want "anonymous" users
  - Users want to talk with unverified servers
  - Users want to run unverified code
- Possible approaches:
  - Verification of identity + trust
    - X.509 certificates
  - Enforcement
    - Java security model

Network Security

- What properties would we like the network to offer?
  - Privacy: messages can't be eavesdropped
  - Integrity: messages can't be tampered with
  - Authenticity: we can verify who created the message
  - Timeliness: we can verify that the packet was sent not too long ago
  - Availability: I can send and receive the packets I want
  - Non-repudiation: you can't claim you didn't say something you did
  - Anonymity: not only can't you tell what the content of my conversation is, you can't even tell who I'm talking with
- There are other properties we would like from the distributed services that run on top, as well
  - E.g., if I send you my medical records, you can't send them to anyone else

Achieving Security

- It's not about making security violations impossible, it's about making them too expensive to be worth it to the attacker
  - Example: There's a simple method to break passwords: try them all
- Security is a negative goal
  - Proof that something can't be done within some cost model is often followed by demonstration that it can be done by stepping outside the model
    - Example: dictionary attacks
      (Goal isn't "break into account gwb," it's "break into any account")
- There is a long-standing debate about the roles of prevention and retaliation
  - Steel plates over your doors and windows or deadbolts and the legal system?

Attack Models

- eavesdropper
- man-in-the-middle
- replay attack
- spoof
- phishing
- …

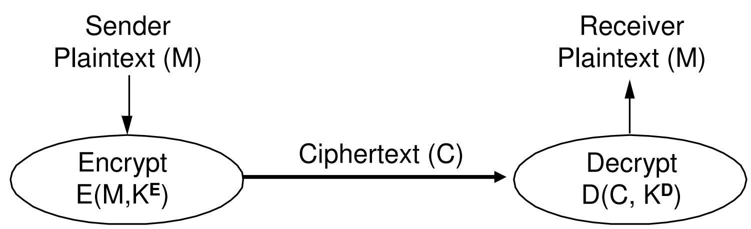
Part I: Privacy/Secrecy

- Main goal: prevent an eavesdropper from understanding what is being sent

Basic Tool: Cryptography

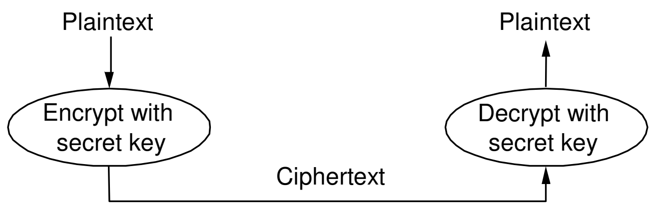
- Cryptography (encryption) directly addresses the eavesdropper problem
- It turns out it can also be used to address some of the other problems
  - E.g., authenticity
- Encryption is a building block
  - A *security protocol* is needed to achieve some more complex goal

Basic Encryption for Privacy



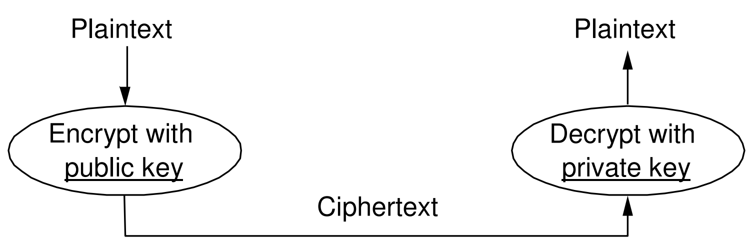
- Cryptographer chooses functions E, D and keys K^E, K^D
  - Mathematical basis
- Cryptanalyst try to “break” the system
  - Depends on what is known: E and D, M and C?

Secret Key Functions (DES, IDEA)



- Also called “shared secret”
- Single key (symmetric) is shared between parties
  - Used both for encryption and decryption
- Pro’s:
  - Fast; hard to break given just ciphertext
- Con’s:
  - key distribution problem
    - Suppose you want to create an account at youTube.com?
- The key distribution problem is crippling
  - Every client must share a (distinct!) secret with every server

Public Key Functions (RSA)



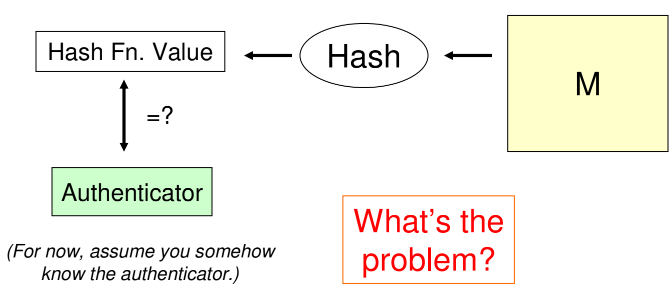
- Public key can be <u>published</u>; private is a secret
  - Still have a key distribution problem, though…

Properties of Public Key Encryption

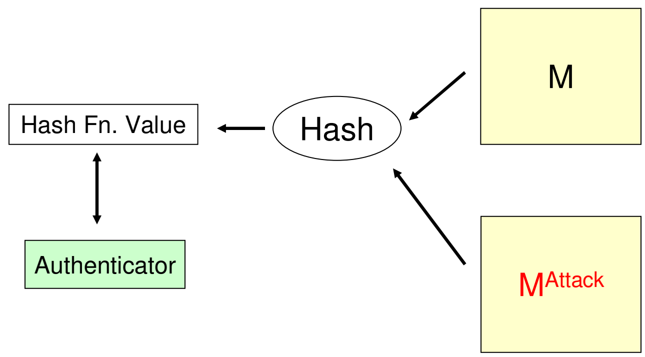
- Let K^1 be the private key, and K^* be the public key
- $D(E(M,K^*), K^1) = M = D(E(M,K^1), K^*)$
- Implications
  - Anonymous client can send private message to server knowing only K^*
  - Server can prove authenticity by encrypting with K^1

Part II: Integrity

- Main goal: detect that a message has been altered
- Main ideas:
  - Redundancy: same idea as checksum



Integrity

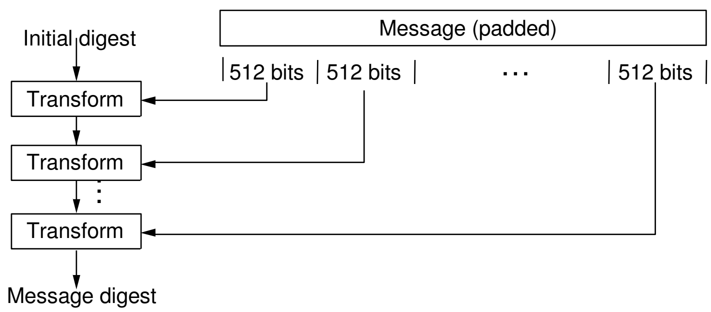


Cryptographic Hash

- Basically:
  - A hash function (maps arbitrary sized data to a fixed number of bits)
  - Given message M, is cheap to compute
  - Give a hash value, it's hard to find data that produces that value
    - Ideally, a change to any one bit of the message flips each bit of the hash value with probability 0.5
- Result:
  - Even if the attacker knows the authenticator value, can't produce bogus data that matches it

Message Digests (MD5, SHA)

- Act as a cryptographic checksum or hash
  - Typically small compared to message (MD5 128 bits)
  - “One-way”: infeasible to find two messages with same digest



Example: Secure File System (SFS)

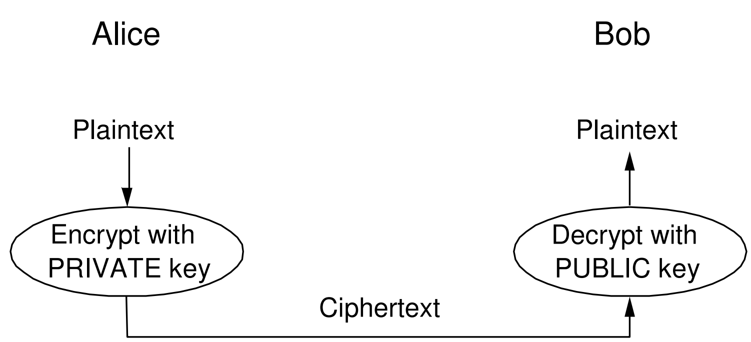
- Goal: use untrusted nodes on web (e.g., your machines) to host a secure file system
  - Main problem: How does someone fetching a file from you know that you’re not returning nonsense?
- Main idea: the “names” of files are cryptographic hash values of their contents
- Directories entries: `[string file name, hash value]`
- When you fetch a file, you can verify that it’s the one you asked for!
- (How do you verify the root directory…?)

Part III: Authenticity (and Integrity)

- Q: How can you verify that a message claiming to be from Alice is actually from Alice?

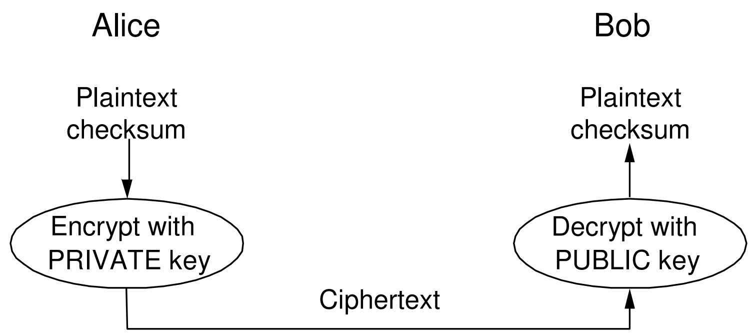
- A: The message proves that the sender knows something that only Alice knows.
  - Primary example: Alice's private key

Basic Idea



Is there a problem?

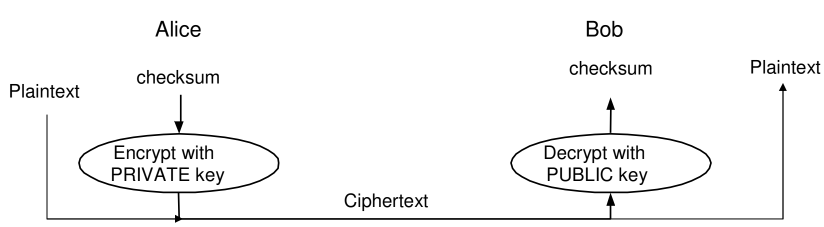
Authenticity + Integrity



CSE/EE 461, Autumn 2007 M15.23

A Faster Version

- Encryption can be expensive, e.g., RSA measured in Kbps
- To speed up, let's sign just the checksum instead!
  - Check that the encrypted bit is a signature of the checksum



- RSA Digital Signature:
  - Use a cryptographic hash
    - Why?

CSE/EE 461, Autumn 2007 M15.24

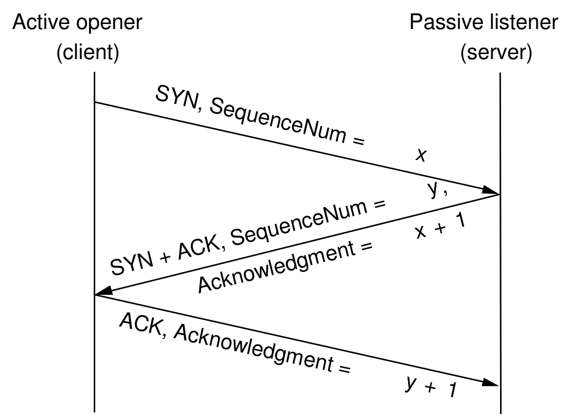
Message Integrity / Authenticity

- Sender:
  - computes cryptographic hash of message M
  - encrypts the hash with its own private key
  - Sends both M and the encrypted hash
- Receiver:
  - Accepts M and the encrypted hash
  - Applies the sender's public key to decrypt the hash
  - Computes the hash on M and compares it to the decrypted value

Part IV: Timeliness

- Want to guard against replay attacks
- Why not just send the time with each message?
- General idea: send a 'nonce'
  - Usually a random number chosen from a large space
  - Responder must reply with an indication they understood this value (e.g., by repeating it)

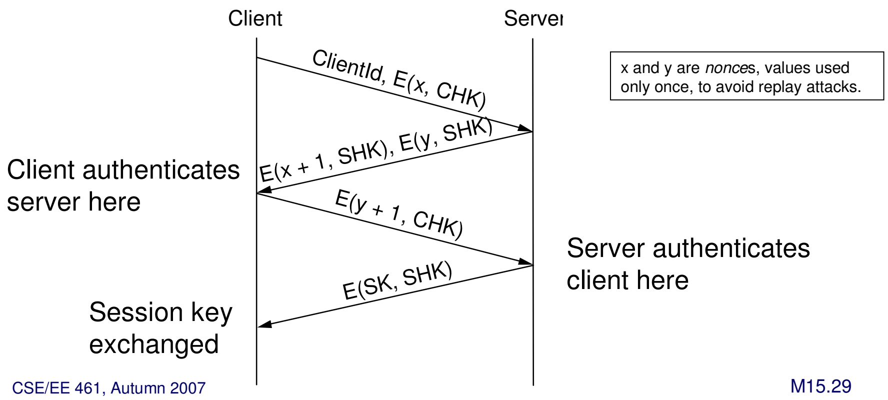
Nonce Example: TCP



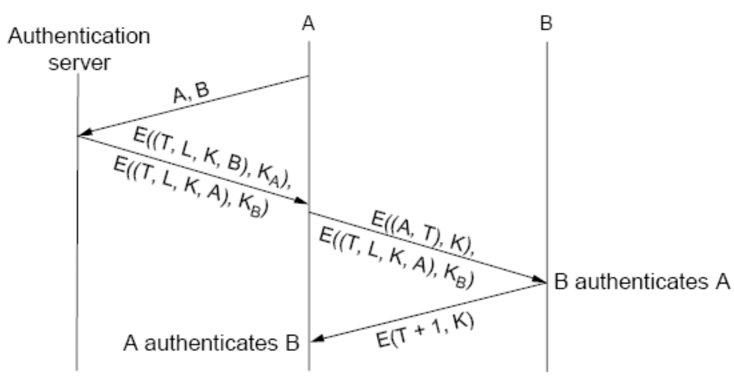
Part V: Security Protocols

Authentication w/ Shared Secret

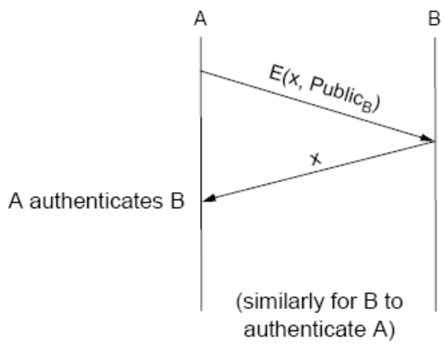
- Three-way handshake for mutual authentication
  - Client and server share secrets, e.g., login password



Via Trusted Third Party (Kerberos)



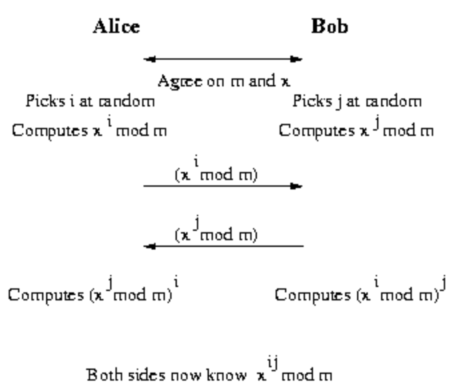
Public Key Authentication



CSE/EE 461, Autumn 2007 M15.31

Diffie-Hellman Key Exchange

- Problem: agree on a session key with no prior information exchanged

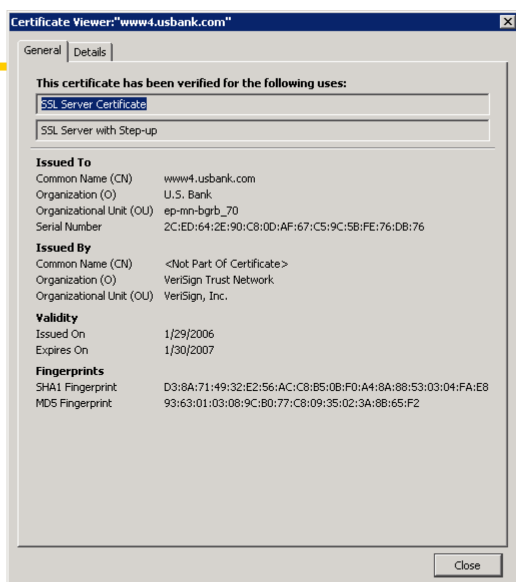


CSE/EE 461, Autumn 2007 M15.32

ssh

- Encrypted channel
  - Diffie-Hellman key exchange (plus negotiated encryption scheme)
- Authentication
  - Client has private key on local machine (usually in `~/.ssh/id_rsa`) and public key on remote machine (in `~/.ssh/authorized_keys)`
  - Server sends a challenge for client to sign using private key
  - Server verifies challenge using public key

X.509 Certificates



Security in Context

- A system is only as secure as its weakest link
- Often that weakest link is you!
- Example: You're a registered user with, say, 25 online services. How many different passwords do you have?
  - Want "single sign-on"
  - Need either:
    - A client-side password manager, or
    - A central, trusted authority *a la* Kerberos (Microsoft Passport, Google Checkout)

Social engineering

- Con person into giving out information
- Phone secretary, say:
  - "Hi. I'm your company's IT administrator. Your boss is currently traveling, and I can't reach them. I need their password to verify their account hasn't been broken into. This is really urgent."
- Somebody phones you, and says:
  - "Hi. I'm with the Bank of America credit card fraud division. We've detected suspicious activity on your account, and we want to ensure you haven't become a victim of identity theft. Before we start, I need to verify your identity. What is your bank account number? SSN?"
- Often far more effective than technical attack
  - requires all people with access to sensitive information to be conscious of security issues

CBS NEWS

BACK | PRINT

Patricia Dunn: I Am Innocent

PALO ALTO, Calif., Oct. 8, 2006

(CBS) The Hewlett-Packard board of directors was a leaky ship. Secret board deliberations were ending up in the press left and right, and it was decided something had to be done.

That something is arguably the most famous leak investigation since Watergate, and because of it Pattie Dunn, who was chairman of the HP board of directors, now faces criminal charges, and could go to jail.

As **correspondent Lesley Stahl** reports, the charges stem from the use of something called pretexting, where phone records are retrieved by subterfuge and pretense – where someone calls the phone company and pretends to be someone else in order to obtain the records.

The tactic was apparently used to retrieve the phone records not only of HP board members but of reporters as well. Social security numbers were also obtained, board members and journalists were followed, and there was even discussion of planting spies in newsrooms.

On Thursday, Pattie Dunn was booked on four felony counts in connection with the investigation.

Microsoft Security Bulletin MS01-017

Erroneous VeriSign-Issued Digital Certificates Pose Spoofing Hazard

Originally posted: March 22, 2001
Updated: June 23, 2003

Summary

Who should read this bulletin:
All customers using Microsoft® products.

Impact of vulnerability:
Attacker could digitally sign code using the name "Microsoft Corporation".

Recommendation:
All customers should install the update discussed below.

Technical description:

In mid-March 2001, VeriSign, Inc., advised Microsoft that on January 29 and 30, 2001, it issued two VeriSign Class 3 code-signing digital certificates to an individual who fraudulently claimed to be a Microsoft employee. The common name assigned to both certificates is "Microsoft Corporation". The ability to sign executable content using keys that purport to belong to Microsoft would clearly be advantageous to an attacker who wished to convince users to allow the content to run.

The certificates could be used to sign programs, ActiveX controls, Office macros, and other executable content. Of these, signed ActiveX controls and Office macros would pose the greatest risk, because the attack scenarios involving them would be the most straightforward. Both ActiveX controls and Word documents can be delivered via either web pages or HTML mails. ActiveX controls can be automatically invoked via script, and Word documents can be automatically opened via script unless the user has applied the Office Document Open Confirmation Tool.


United States | Microsoft.com Hor

Microsoft

Search Microsoft.c

Help and Support

Help and Support Home | Select a Product | Search Knowledge Base

Update Available to Revoke Fraudulent Microsoft Certificates Issued by VeriSign

View products that this article applies to.

This article was previously published under Q293811

Article ID : 293811
Last Review : October 27, 2006
Revision : 3.3

On This Page

- SUMMARY
  - Important Notes
- MORE INFORMATION

SUMMARY

In March, 2001, VeriSign, Inc. announced that it had issued two digital certificates to an individual who fraudulently claimed to be a Microsoft employee. This issue is discussed at length in Microsoft Security Bulletin MS01-017. VeriSign has revoked these certificates, and they are listed in the current VeriSign Certificate Revocation List (CRL). However, because the VeriSign code-signing certificates do not specify a CRL Distribution Point (CDP), it is not possible for any browser's CRL-checking mechanism to locate and use the VeriSign CRL. Microsoft has developed an update that rectifies this problem. The update package includes a CRL that contains the two certificates, and an installable revocation handler that consults the CRL on the local computer, rather than attempting to use the CDP mechanism.

What is Denial of Service?

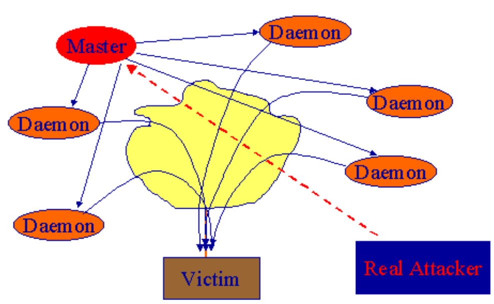
- Attacker can deny service to legitimate users if they can overwhelm the system providing the service
  - System is full of bugs … just send it packets that trigger them
  - System has limited bandwidth, CPU, memory, etc. … just sent it too many packets to handle
- Big issue in practice and lack of effective solutions
  - Today, patch as found (CERT) or build implementation to tolerate DOS
  - Tomorrow, design protocols to withstand, possibly network support for shutting down attack?
- Two broad classes:
  - Nasty packets trigger implementation bugs, e.g., Ping of Death
  - Packet floods target bandwidth, CPU, memory, e.g., SYN flood

Complication: Spoofed Addresses

- Why reveal your real address? Instead, “spoof” it.
  - Can implicate others and appear to be many hosts
- Solution?
  - Ingress filtering (ISPs check validity of source addresses) helps, but has poor incentive patterns and is not a complete solution
- Opportunity: “backscatter analysis”
  - host responds to spoofed packet, sends response packet to essentially random IP
  - if you have a large number of unused IPs, just listen and you’ll hear the backscatter -- can measure DOS attacks!

Distributed DOS (DDOS)

- Use automated tools to set up a network of zombies
  - Trin00, TFN, mstream, Stacheldraht, …



Operation Bot Roast

HOME PAGE | MY TIMES | TODAY'S PAPER | VIDEO | MOST POPULAR | TIMES TOPICS

The New York Times

Technology

WORLD | U.S. | N.Y. / REGION | BUSINESS | TECHNOLOGY | SCIENCE | HEALTH | SPORTS | OPINION

Search Tech News & 8,000+ Products

Browse Products

-- Select a Product Category --

Police swoop in on New Zealand botmaster

By LIAM TUNG, FOR ZDNET AUSTRALIA
Published: November 30, 2007

New Zealand Police this week cracked down on an alleged botnet ringleader in New Zealand, who the FBI claims had illegal control over 1 million computers.

SIGN IN TO E-MAIL OR SAVE THIS

PRINT

ARTICLE TOOLS SPONSORED BY JUNO

CNET News.com

More resources from CNET:

- More Tech News
- Download Free Shareware
- Find Product Reviews
- Compare Product Prices

Search CNET for:

The sweep is part of the FBI's second phase of "Operation Bot Roast"--the same operation which resulted in four felony charges against 26-year-old Los Angeles security consultant John Schiefer.

Lessons

- Encryption is powerful tool
  - strong mathematical properties
  - used to provide integrity, authenticity, privacy
  - must be used correctly
- Many other security issues in practice
  - non-mathematical, “best practice” based
  - easy to get wrong
- In the end, people are the weak link
  - social engineering