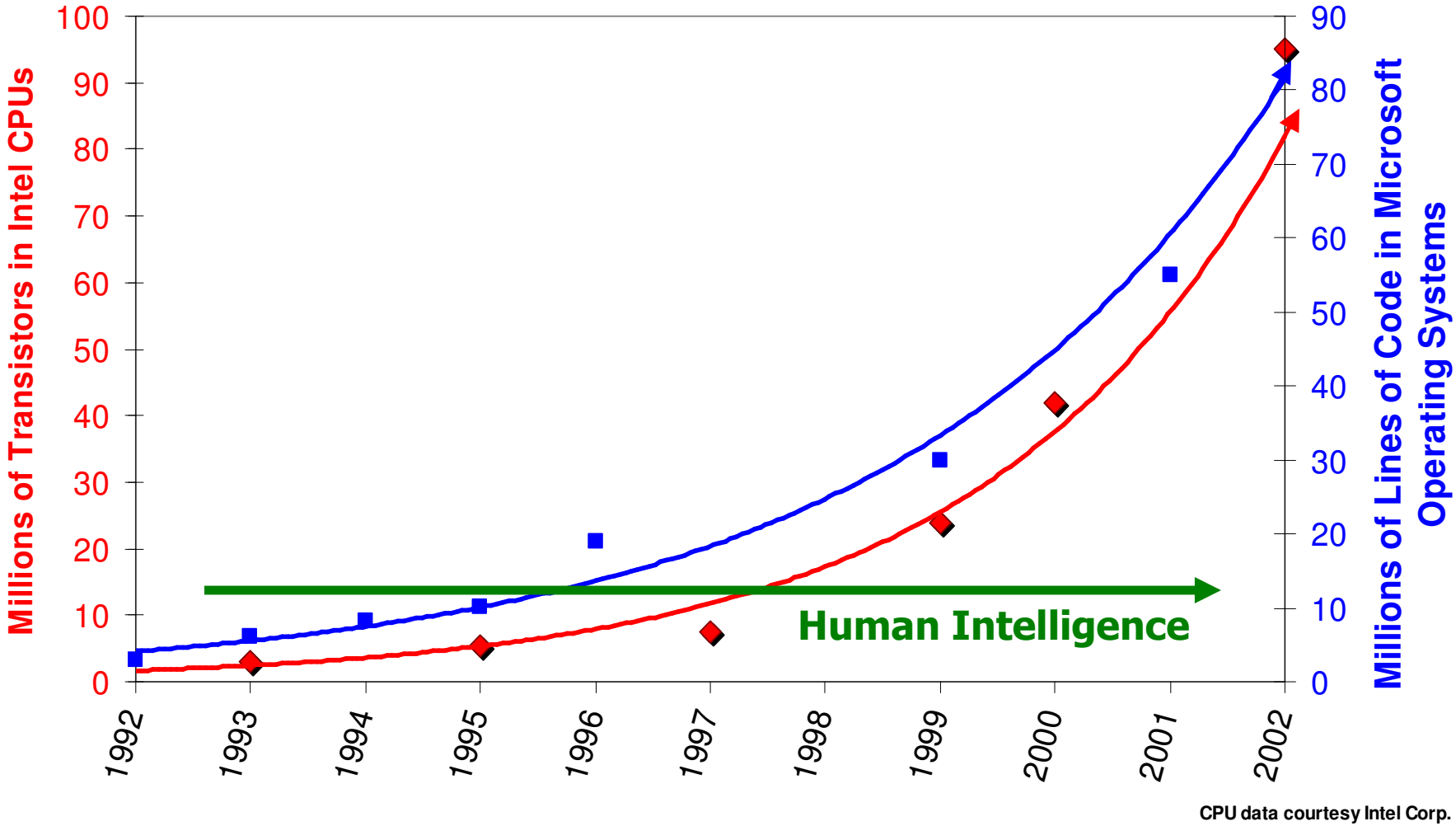

# Beyond Applied Cryptography: Designing a Secure Application

Session #66

MIS TRAINING INSTITUTE

CRYPTOGRAPHY RESEARCH

# Managing Complexity




CPU data courtesy Intel Corp.

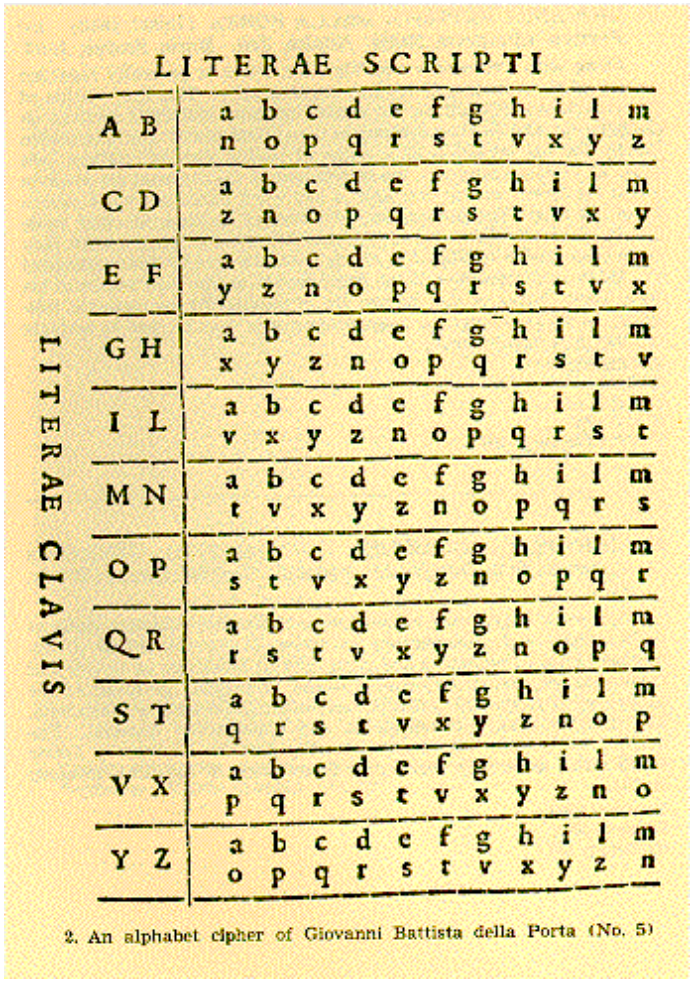
# About Cryptography Research

- Solve difficult real-world problems
  - Systems designed by CRI engineers protect >$50B annually
  - Industries: Financial, content, pay TV, communications
- Specialties
  - Tamper-resistance, content security, platform security

# About Nate

- Designs network security products
  - RealSecure
  - Storage Appliances
  - NAT TCP splicer, layer2 IPSEC, etc.
- Companies: ISS, InfoGard, Decru
- FreeBSD
  - ACPI, Storage

# Cryptography in 3 Slides



LITERAE SCRIPTI

LITERAE CLAVIS

| | a | b | c | d | e | f | g | h | i | l | m |
|---|---|---|---|---|---|---|---|---|---|---|---|
| A B | n | o | p | q | r | s | t | v | x | y | z |
| C D | a | b | c | d | e | f | g | h | i | l | m |
| | z | n | o | p | q | r | s | t | v | x | y |
| E F | a | b | c | d | e | f | g | h | i | l | m |
| | y | z | n | o | p | q | r | s | t | v | x |
| G H | a | b | c | d | e | f | g | h | i | l | m |
| | x | y | z | n | o | p | q | r | s | t | v |
| I L | a | b | c | d | e | f | g | h | i | l | m |
| | v | x | y | z | n | o | p | q | r | s | t |
| M N | a | b | c | d | e | f | g | h | i | l | m |
| | t | v | x | y | z | n | o | p | q | r | s |
| O P | a | b | c | d | e | f | g | h | i | l | m |
| | s | t | v | x | y | z | n | o | p | q | r |
| Q R | a | b | c | d | e | f | g | h | i | l | m |
| | r | s | t | v | x | y | z | n | o | p | q |
| S T | a | b | c | d | e | f | g | h | i | l | m |
| | q | r | s | t | v | x | y | z | n | o | p |
| V X | a | b | c | d | e | f | g | h | i | l | m |
| | p | q | r | s | t | v | x | y | z | n | o |
| Y Z | a | b | c | d | e | f | g | h | i | l | m |
| | o | p | q | r | s | t | v | x | y | z | n |

2. An alphabet cipher of Giovanni Battista della Porta (No. 5)

# 1. Encryption Is Not Security

- Many systems use encryption
  - 802.11b WEP
  - DVD CSS
- Failures result of misuse
  - Constant re-keying
  - Poor key management
- Neither fell first to brute force of key

# 2. Strength Is Not Assurance

- Strength
  - Resistance to Known Attacks
  - Example: key lengths
- Assurance
  - Likelihood to Fail to Unknown Attacks
  - Example: SSL 3.0 hash/keying

# 3. Use standards

- Good standards last
  - SSL 3.0 (1996)
  - SHA-1 (1996)
  - DES (1977)
- New crypto suffers 50% mortality rate

# What doesn't work

- Committee designs

# What doesn't work

- Obscurity
  - Increases cost for initial attack, but not repeat attacks
  - Reduces relying party's ability to gain assurance

# What doesn't work

- Fixed certification standards
  - Standardized evaluations only catch standardized attacks…

GRADUATION CHECKLIST FOR MCDB MAJORS

Required Major Courses (Grades must be C- or better, including MCDB electives. See page 7 for details.)

| ✓ | Course # | Course Name | All variations (AP, testing, transfer, etc.) |
|---|---|---|---|
| ___ | MCDB 1150 | Introduction to Cell and Molecular Biology | _____ |
| ___ | MCDB 1151 | Introduction to Cell and Molecular Biology Lab | _____ |
| ___ | MCDB 2150 | Principles of Genetics | _____ |
| ___ | MCDB 2151 | Principles of Genetics Laboratory | _____ |
| ___ | MCDB 3100 | Cell Biology | _____ |
| ___ | MCDB 3140 | Cell Biology Laboratory | _____ |
| ___ | MCDB 3650 | Molecular Biology | _____ |
| ___ | MCDB 4620 or 4680 | Developmental Biology | _____ |
| ___ | MCDB 4622 or 4660 | Developmental Biology Laboratory | _____ |

MCDB Electives (Minimum of 5 hrs. required. 3 hrs. total, 3 hrs of approved non-MCDB courses may also be used)

| | | | |
|---|---|---|---|
| ___ | MCDB | ____ | _____ |
| ___ | MCDB | ____ | _____ |
| ___ | ____ | ____ | _____ |

MCDB Ancillary Requirements in CHEM, MATH, PHYS (See page 7 for minimum grade requirements.)

| | | | |
|---|---|---|---|
| ___ | CHEM 1111 | General Chemistry I | _____ |
| ___ | CHEM 1121 | General Chemistry II | _____ |
| ___ | CHEM 3311 | Organic Chemistry I | _____ |
| ___ | CHEM 3321 | Organic Chemistry Laboratory I | _____ |
| ___ | CHEM 3331 | Organic Chemistry II | _____ |
| ___ | CHEM 3341 | Organic Chemistry Laboratory II | _____ |
| ___ | CHEM 4711 | General Biochemistry I | _____ |
| ___ | MATH 1300 | Calculus and Analytical Geometry I | _____ |
| ___ | PHYS 1110 or 2010 | General Physics I | _____ |
| ___ | PHYS 1120 or 2020 | General Physics II | _____ |
| ___ | PHYS 1140 or 2020 | General Physics Laboratory | _____ |

Core Curriculum Requirements not Satisfied by MCDB Major or Ancillary Courses (D- or better grades)

| ✓ | Requirement | Course # | Comments: AP, Test, Transfer, etc. |
|---|---|---|---|
| ___ | Written Communication (lower division) | _____ | _____ |
| ___ | Historical Context | _____ | _____ |
| ___ | Culture and Gender Diversity | _____ | _____ |
| ___ | United States Context | _____ | _____ |
| ___ | Literature and the Arts (lower division) | _____ | _____ |
| ___ | Contemporary Societies | _____ | _____ |
| ___ | Ideals and Values | _____ | _____ |
| ___ | Literature and the Arts (upper division) | _____ | _____ |
| ___ | Written Communication (upper division) | _____ | _____ |
| ___ | Critical Thinking | _____ | _____ |

MAPS, Language, Other Items not Covered Above

| ✓ | Requirement | Course # | Comments |
|---|---|---|---|
| ___ | _____ | _____ | _____ |
| ___ | _____ | _____ | _____ |

Additional Upper Division Credits Needed to Bring Total to 45

| ✓ | Course # | Hours | Course Name & Comments |
|---|---|---|---|
| ___ | _____ | ___ | _____ |
| ___ | _____ | ___ | _____ |
| ___ | _____ | ___ | _____ |

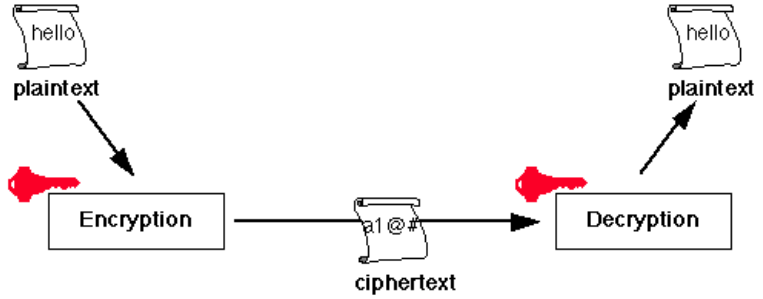
___ Total Semester Hours ≥ 120 ___ Last 30 hours in residence at CU Boulder

___ Overall GPA ≥ 2.0 ___ MCDB GPA ≥ 2.0 ___ Ancillary GPA ≥ 2.0

# What doesn't work

- Requirements that go against security
  - More speed, more features, less cost, less development time
- "Hail Mary" security evaluations
  - Too late:  Need security by design

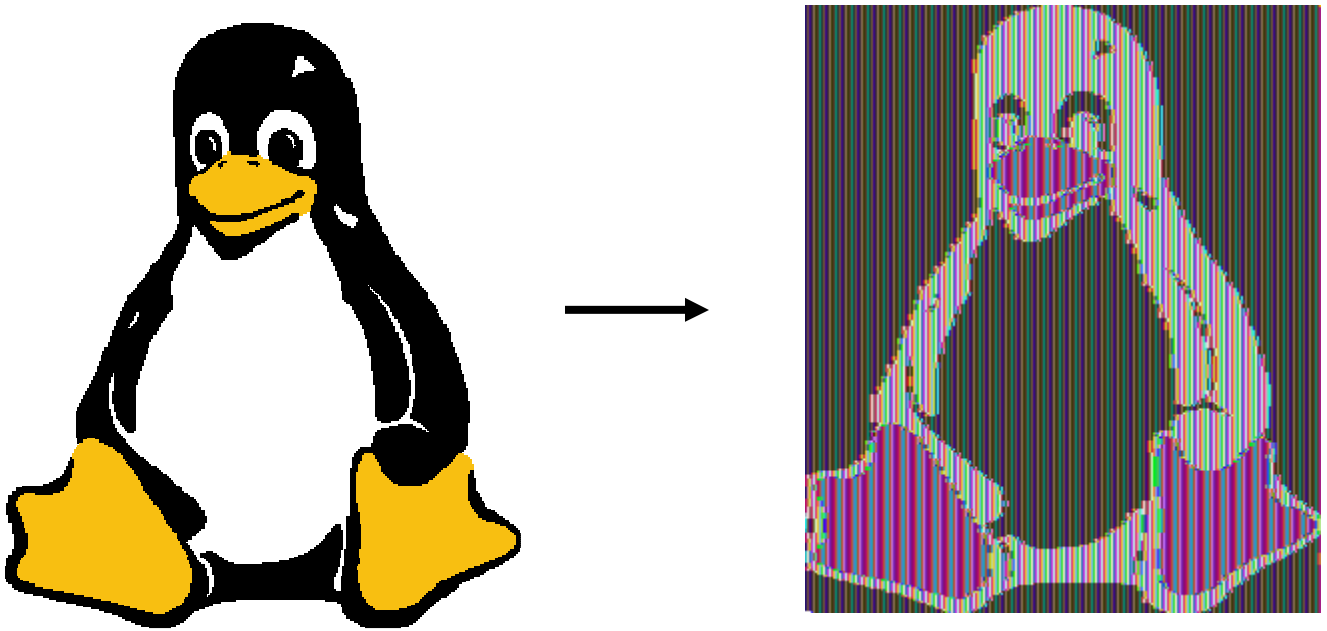
# Example: Crypto Storage Product




1. "We'll encrypt the data."
2. Read crypto book
3. Argue about algorithm choice
4. Argue about key length
5. Implement using downloaded code
6. Test
    1. Encrypted data looks "random"
    2. Data decrypts correctly
7. Ship!

# Failure: Cipher Mode

- Customer generates secret image
- Encrypts it with new product
- Result?



# What went wrong?

- Focused on unlikely attacks
  - Algorithm
  - Key Length
- Overlooked likely attacks
  - Improper cipher mode
  - Others: RNG, key management, side-channel leakage, etc.

# Actual Risk vs. Perceived Risk

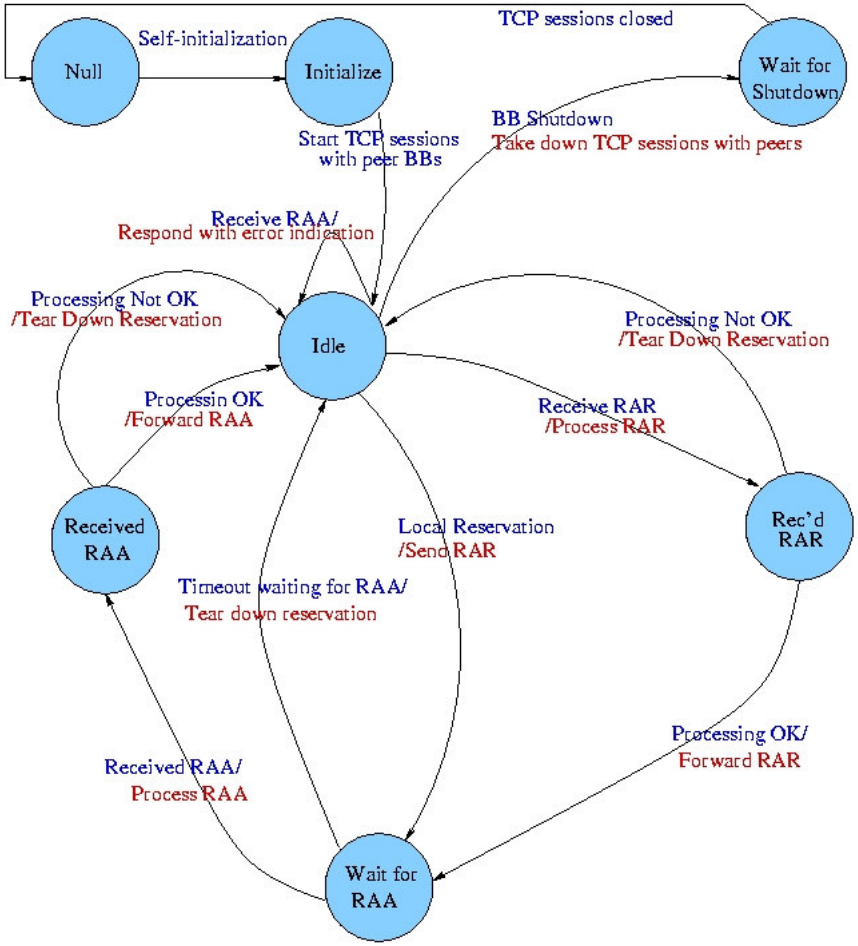
- Real Quote:
  - "Smart cards with triple DES are three times as secure as those using single DES."

Everybody knows this is wrong…

Attackers almost never waste time and money on brute force

Even when it's easy, there are easier attacks

# Protocol Analysis



# Simple Protocol Analysis

- On three big pieces of paper…
  - ① Chart the protocol flow
    - Include every message that can be sent
    - Error messages, optional messages, etc.
  - ② List what can be discovered about each cryptographic value
    - Each crypto step generally reveals something new
    - List everything (helps catch unintended interactions)
  - ③ Diagram the state machine of each participant
    - Include negotiated options, failure states, crypto, etc.
- Reconcile possible end states against objectives.

# Common Protocol Weak Spots

- Algorithm negotiation
- Version negotiation (backward + forward)
- Man-in-the-middle
- Message replay (within a session, multiple sessions)
- Message forwarding & impersonation
  - A connects to B, who connects to C pretending to be A
- Certificate handling & validation (or lack thereof)
- Out-of-sequence messages
- Error handling reveals information
- Denial of service
- Timing analysis
- Excessive complexity or lack of defined state machine
- Improper or inadequate use of hash functions
- Inefficiencies (round trips)
- Redundant information
- Management/debug functions (code upgrades, etc.)

# 10 Suggestions

- Goal is higher assurance that system meets security requirements
- Security is absence of functionality

- This is hard…

10 Suggestions

❶ **View security in economic terms.**

Assign a dollar-value to your risk.

Get management support for the estimate.

Spend before problems get out of control.

# 10 Suggestions

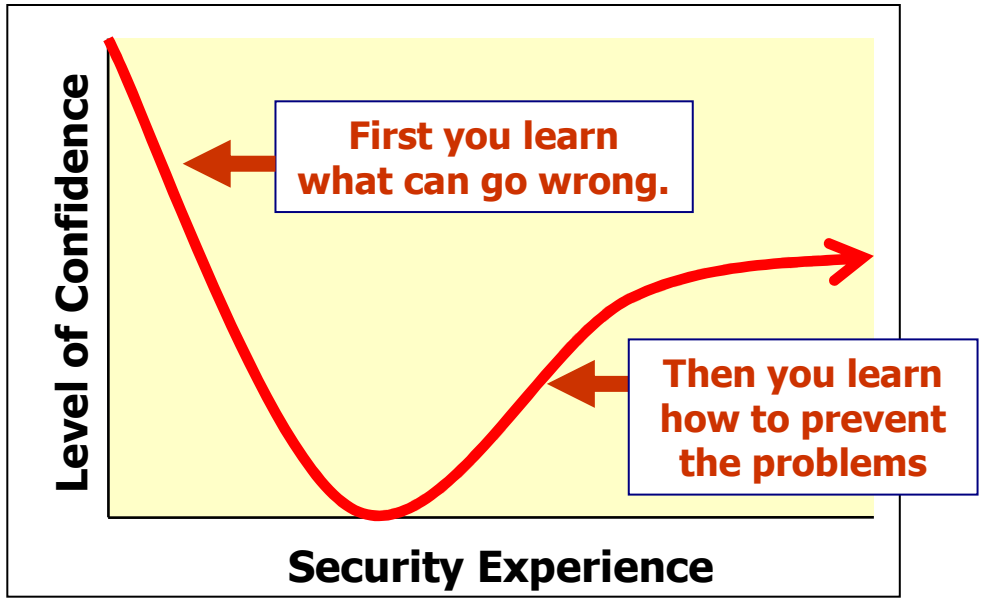
❶ View security in economic terms.

❷ **Think about how risk is allocated.**

- Where are the single points of failure?
- Will those who control your risk share it?
- Are the people you trust actually trustworthy?
  - What is their historical track record?
  - Do they make unsubstantiated claims of “security”?
  - Who can vouch for their work?

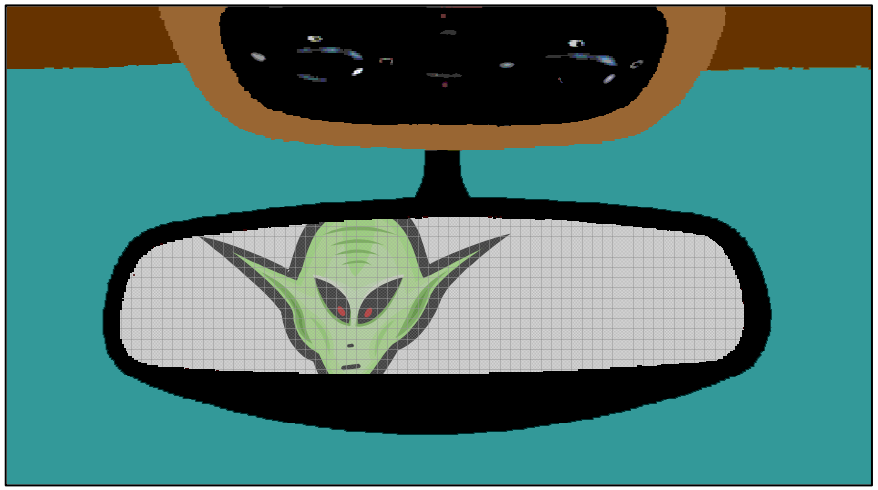
# 10 Suggestions

1. View security in economic terms.
2. Think about how risk is allocated.
3. **Be humble and know your limits.**
   - Don't mistake confidence for experience.
   - Encourage people to look for flaws in your work.
   - Don't assume attackers won't "figure it out".



# 10 Suggestions

1. View security in economic terms.
2. Think about how risk is allocated.
3. Be humble and know your limits.
4. **Make realistic assumptions.**
   - Assume that users are lazy and gullible.
   - Assume that engineers make mistakes.
   - Beware of the rear view mirror.
     - Your greatest risk may not be what went wrong last time.



## 10 Suggestions

1. View security in economic terms.
2. Think about how risk is allocated.
3. Be humble and know your limits.
4. Make realistic assumptions.
5. **Minimize complexity.**
   - Isolate critical components.
   - Beware of complex interfaces.
   - Have the courage to resist adding features.

Complexity is a security flaw.

# 10 Suggestions

1. View security in economic terms.
2. Think about how risk is allocated.
3. Be humble and know your limits.
4. Make realistic assumptions.
5. Minimize complexity.
6. **Spend more on evaluation than design.**
   - Evaluations can only prove insecurity.
   - Make sure evaluators are skilled and objective.
     - Don't impose unreasonable restrictions.
     - Requires creativity, experience, attention to detail.

# 10 Suggestions

1. View security in economic terms.
2. Think about how risk is allocated.
3. Be humble and know your limits.
4. Make realistic assumptions.
5. Minimize complexity.
6. Spend more on evaluation than design.
7. **Be a skeptic.**
   - Assume systems are insecure unless you have evidence to the contrary.
     - Avoid anything undocumented or untestable.
   - Ask tough questions and demand responses.
     - Don't be impressed by the line:
       - "We can't tell you for security reasons."

# 10 Suggestions

1. View security in economic terms.
2. Think about how risk is allocated.
3. Be humble and know your limits.
4. Make realistic assumptions.
5. Minimize complexity.
6. Spend more on evaluation than design.
7. Be a skeptic.
8. **Plan for trouble.**
   - What happens after a breach?
     - Will you know if there was a breach?
   - Keep good audit records.
   - Are “impossible” attacks really impossible?

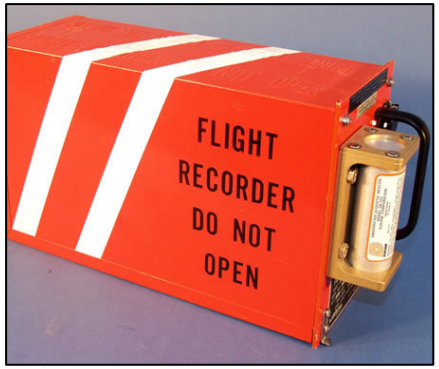



Image courtesy NTSB.

# 10 Suggestions

1. View security in economic terms.
2. Think about how risk is allocated.
3. Be humble and know your limits.
4. Make realistic assumptions.
5. Minimize complexity.
6. Spend more on evaluation than design.
7. Be a skeptic.
8. Plan for trouble.
9. **Use both internal & external expertise.**

   Risks are much higher if you rely only on just one.
   Get multiple opinions, especially if you fear:
   piracy, fraud, or espionage.

# 10 Suggestions

1. View security in economic terms.
2. Think about how risk is allocated.
3. Be humble and know your limits.
4. Make realistic assumptions.
5. Minimize complexity.
6. Spend more on evaluation than design.
7. Be a skeptic.
8. Plan for trouble.
9. Use both internal & external expertise.
10. **Study all layers of the system**
    Transistors up to business objectives

# Unsolved Problems

- Why does the future take so long to appear?
- Why is it so incomplete when it does?

Some unsolved problems that really bug me…

# Unsolved: Application Filtering

- Progression of the firewall
  - IP address, port filtering
  - Stateful inspection (Latest: IPS)
  - Application proxy

# Unsolved: Application Filtering

- Protocols are multiplying
- Firewalls stopped evolving at the HTTP layer
  - Everything runs on port 80
  - Protocols change greatly between versions
  - “All or nothing” filtering

# Unsolved: Application Filtering

- Proposal: developer tools output protocol specification
  - Client and server software use spec to format messages
  - Firewall uses protocol spec to disallow improper messages

# Unsolved: Certification Aging

- Certification specifies testing conditions
  - FIPS 140: certificate shows tested configuration
  - Common Criteria: profile lists requirements

# Unsolved: Certification Aging

- Problem: No one has that exact configuration and soon the vendor releases a new version
  - Windows NT level 2 cert (but without network)
  - Netscape level 2 cert (but you need a sticker)

# Unsolved: Certification Aging

- Proposal: Use a questionnaire
  - Vendors answer standardized questions about their system
  - Customers (along with evaluators) use the answers to identify application-specific questions
  - Vendor publishes the results

# Conclusions

When designing or evaluating a secure application remember:

1. Encryption is not security
2. Strength is not assurance
3. Use standards

If in doubt, call an expert…

# Contact Information

For more information, or to discuss how Cryptography Research can help with a security problem:

Nate Lawson
nate@cryptography.com
www.cryptography.com

Questions?

# Single Points of Failure

Examples

| | | |
|---|---|---|
| ROM/E$^2$/BIOS contents | Hard disk controllers | Data backup & redundancy |
| Key storage & metadata | Revocation systems | Crypto algorithms |
| Threat detection systems | Engineering personnel | Drivers |
| Executable program storage | Compiler correctness | CPU execution correctness |
| Sandboxes | Non-standboxed code | Security protocols |
| Input validation routines | Passwords & login procedures | Tamper resistance |
| Software update procedures | Master keys & passwords | |

# Trust Boundaries

- Designers should isolate keys & critical components
  - Putting all your eggs in one basket is actually good
    - Risking all your eggs in many baskets is dangerous.
    - Fewer critical components means they can be tested better.
- Most products have poorly-defined boundaries.
  - Are the perimeters (or contents) too complex?
    - Typical Windows PC is too complex to secure internally.
  - What can cross the perimeter?
    - APIs, network protocols, chip I/Fs, control/audit/backup data…
  - Analyze single points of failure (inside & outside)  [next]

# Tools

- Gathering Information
  - Crypto toolkits (Crypto++, CryptoLib, etc.)
  - Statistical toolkits (custom)
  - Bignum libraries (NTL for Lattice Reduction)
  - Compiler, system analysis tools, debugger, decompiler
  - Network traffic recorder (tcpdump)
- Brute force / disaster recovery
  - FPGA board, CPU farm
  - Password dictionaries
  - Hard drive imaging tools
  - Password recovery tools/services
- Tamper Resistance
  - DPA workstation
  - Oscilloscope
  - X-ray, Probe station, microscopes, e-beam, FIB