

Distributed Computing: Chapter 7 & 8 -- case study: Cryptography Policy

Doug Tygar
IS 206
16 September 1999

Cipher system

- Encryption function
 $\text{ciphertext} = E(\text{plaintext}, \text{e-key})$
- Decryption function
 $\text{plaintext} = D(\text{ciphertext}, \text{d-key})$
- Private-key (symmetric cryptography)
 $\text{e-key} = \text{d-key}$
- Public-key (asymmetric cryptography)
 $\text{e-key} \neq \text{d-key}$
and d-key is not easily derivable from e-key

Signatures in public key crypto

- To send a secure message
 - keep d-key secret
 - make e-key public
- To manage keys
 - send a key under secret e-key
 - this can be used as private (symmetric) key for private key cryptography
 - usually private key cryptography is much faster (typically 3-5 orders of magnitude) than public key cryptography with same security level
 - To sign a message
 - make d-key public
 - keep e-key secret

Public key cryptography in media

- From *Sunward Journey* by Katherine Coffaro (*American Harlequin Romance*):

"I'm not really into computers, Jay. I don't know much. I do know the key to the code was the product of two long prime numbers each about a hundred digits, right?"

"Yes, that's correct. It's called the RSA cryptosystem"

"Right, for Rivest, Shamir and Adleman from MIT. That much I know. I also understand that even using a sophisticated computer to decipher the code it would take forever," she recalled. "Something like three point eight billion years for a two-hundred digit key, right?"

"That's exactly correct. All the stolen information was apparently tapped from the phone lines running from the company offices to your house.

Supposedly no one except Mike had the decoding key, and no one could figure it out unless he passed it along, but there has to be a bug in that logic somewhere" he said, loosening his dark green silk tie. "Vee, it's much warmer than I thought. Would you mind if I removed my jacket?"

"Of course not. You're so formal," she remarked

Why is the Internet secure yet?

We want to:

- Know for sure who we are talking to
- Know the message hasn't been tampered with en route
- Be able to send secrets over the network without the bad guys reading them

The bad guys can:

- See what we transmit
- Modify what we transmit
- Send messages from bogus addresses

What don't we know how to do

- We got crypto
- We got protocols
- We got standards
- Who could ask for anything more?

How it works

- Everyone uses RSA or a similar public key cryptosystem
- Each user has a digitally signed certificate vouching for his (**name, public key**), signed by a **CA** (certificate authority)
- Names should be hierarchical, like tygar@berkeley.edu
- Each organization should manage its own CA
- Can have a tree in an organization, like tygar@cs.berkeley.edu
- Extranets use cross certificates
- Organizations can compete later for Root service — we don't need to pick an organization now

Secure communication

- Once you (reliably) know someone's public key, it's straightforward to have them prove they know the private key (we'll talk about authentication in a future lecture)
- Then you establish a "session key", and you can do integrity protection and/or encryption
- So what's the problem?

Top 10 Reasons Internet is not yet secure

10. Users

• From Kaufman, Perlman and Spencer:

Humans are incapable of storing high-quality cryptographic keys, and they have unacceptable speed and accuracy when performing cryptographic operations. They are also large, expensive to maintain, difficult to manage, and they pollute the environment. It is astonishing that these devices continue to be manufactured and deployed. But they are sufficiently pervasive that we must design our protocols around their limitations.

They want fancy features

- They know how to click on "OK"
- Software is providing all sorts of ways of sending along executables, for useless frills
- Users prefer convenience to security (until they've been personally burned)

9. Jim Bidzos

We started out negotiating an RSA license and decided it would be easier to buy the company

John Adams
(Security Dynamics, Inc)

8. Committees

- To hard to get documentation
- Too many pieces
- Too hard to understand
- No rules
- No convergence — always another one+
- Bitter "mine" vs "yours" battles
- Democracy's really a drag

7. Researchers

- Better off if things are complicated — open-ended opportunities for papers, etc
- No particular motivation to get things deployed
- Explore the fringes of bizarre paranoia, and the world waits for these to get solved before deploying anything — 97% solution would work just fine
- Think most elegant solution is general purpose programming language (like Postscript), so users don't even know when they're doing executables

6. Organizations that want to be Root

- Embed their public key in software, hardware
- The world will forever be dependent on them
- They can have a monopoly — charge whatever they want
- It is not necessary to do things this way:

“Bottom-up” trust model

- Start with your public key as the one you know for sure
- Navigate upward in the naming tree (child vouches for parent, etc.) So there are “up”, “down”, and “cross” links
- Hook orgs together with cross links
- Let the customers choose their own Roots in the future — Roots compete on price, policy
- If your organization manages its CAs well, your fate is in your own hands
- Trust fewest things
- Key rollover easy
- Possible to recover from compromised key
- Compatible with top-down

So why don't people do it this way?

- Confusion — liability issues, wizard factor
- Arm twisting on patent licensing
- Installed base

5. Microsoft

- Quote from Steve Crocker
- Three milestones in OS security:
 - Multics
 - Unix
 - DOS

4. Orange Book

- A checklist of features and grades of security (ratings from D to A1)
- Elaborate and expensive mechanisms to prevent leaking data, even by dishonest users
- Much less protection for data getting destroyed
- Designed for the world of one mainframe, only super trusted personnel have physical access
- But separate computers cheaper, more secure
- “Network” assumed to be “trusted backplane”
- Customers don't want to understand security, think stamp of approval means they're secure
- Companies spend incredible amounts of money getting stamp of approval instead of making things secure

3. Lawyers (in Utah and DC)

- Obfuscation
- Liability issues — mystifies notion of signing
- Makes it dangerous to attempt to provide security without 100% solving everything
- Patents
- Export policies

2. France

- I don't mean to imply France is the only repressive totalitarian state
- Usage controls on crypto by their citizens
- Means even if export controls go away, your product has to work differently in France
- Typical workaround:
 - good crypto for US version
 - 40-bit keys for export
 - no security in France

1. US Government

- Who's kidding who — it's not about export
- Crypto isn't a US secret — easy to implement
- They don't tell us what the rules are
- They don't tell us whether something will be exportable until we build it
- The rules can change at any time
- Since no laws have passed yet restricting crypto in US
 - export laws: make two versions difficult or expensive
 - bureaucracy: easier to just give up on security
 - no export if system internetworks with something gov't can't crack
 - twisting Allies arms

The Rules

(as transcribed and encoded by Ellen Fein and Sherrie Schneider)

1. Be "creature unlike any other."
2. Don't talk to a man first (and don't ask him to dance).
3. Don't meet him halfway.
4. Don't call him and rarely return his phone calls.
5. Don't see him more than once or twice a week.
6. No more than casual kissing on the first date.
7. Don't tell him what to do.
8. Don't open up too fast.
9. Do the Rules even when your friends or parents (= courts & US population) think you're nuts!
10. Do the Rules and you'll live happily ever after.
11. Love only those who love you.

Proposed New Amendment

Congress shall make no law restricting the size of numbers which can be multiplied together; or the number of times by which a number be multiplied by itself; or the modulus by which a number be reduced.