Copyright 2000, S.W. Smith

Dartmouth College
Institute for Security and Technology Studies
Department of Computer Science

www.cs.dartmouth.edu/~pkilab

Dartmouth PKILab:

Trust and Security in Inter-networked Communities
The Dartmouth PKI Lab

Who:
Ed Feustel: ISTS
David Nicol, Sean Smith: CS/ISTS
Robert Brentrup, Larry Levine, William Taylor: Kiewit
Shan Jiang, Sean Richardson, Eileen Zishuang Ye: CS students

What:
Chartered, seed funding by Internet2
Support from IBM Research; others pending...
In order to use public-key cryptography
to solve emerging trust and security problems.

Mission:
The tools that use it
- the cryptographic components
- we need to finish the infrastructure:
  - the cryptographic components
  - to solve emerging trust and security problems.

* = new blood, from gov’t, industry
What is public-key cryptography?

Why is it important?

What are the missing pieces?

Goal: solving real problems • proving research concepts with real prototypes,
The key embodies the privilege to do a transformation.

Public-Key Cryptography
Symmetric Key Cryptography: the privileges are the same.

The key embodies the privilege to do a transformation.

Public-Key Cryptography
Public Key Cryptography: the privileges are different.

Symmetric Key Cryptography: the privileges are the same.

The key embodies the privilege to do a transformation.

- Alice can make one of the keys public
- ...and keep the other one private

Public Key Cryptography

Symmetric Key Cryptography

Plaintext

Encryption

Decryption

Ciphertext

Public Key Cryptography
Encryption

Bob can send a secret message to Alice...

Encryption using Public Key Cryptography

- even though they’ve never met, and share no secrets...
Signatures

- Even though they’ve never met, and share no secrets

- Bob can verify that Alice “signed” a message...
Why is public-key important?

In emerging inter-networked world...

need to electronically express complex information

vast number of communities, individuals, entities

need to electronically express complex information

construct and verify a chain of signed assertions

...with a public-key approach

Z’s admin needs to temporarily enroll delegated employees of X

Contrast a password-based approach...

between X and Z:

database, for the next 3 weeks, because of a joint investigation

and am thus allowed to see certain records in this Z Agency

Example:

"I’m Alice, an employee of X Police Dept, working on Project P,

vast number of communities, individuals, entities

In emerging inter-networked world..."
Authorization

How do we deal with multiple communities, bridging, multiple properties (attributes), complex expression (delegation)?

Application

How can applications engage in PKI-based trust judgments?

Client

How can user tools enable PKI-based trust?

Server

How do we establish foundation for this trust, when computation is vulnerable to insider attack?
Problem: Users need many pieces of information to make a trust judgment at clients.

Example: How can Alice securely connect to the right server for this sensitive X-Z collaboration?

Understanding

Current browsers only use one piece, but neither securely nor
PK gives tools to express this information
Users need many pieces of information to make a trust judgment

Trust Judgment at Clients
Our Approach: build a better browser
that obtains the right information
displays it in an understandable way
that cannot be subverted by hostile web content
attributes that server or content offered?
name of server, or content offered?
what are precise semantics of "independent windows"?
can attribute calculation be remote?
what about nested windows?
then what about nested windows?
what about the reality of "crowded rooms"?

Research Issues:

...explore next step...

Plan:
formalize, evaluate planned semantics
prototype UI’s for these
...explorenext step...

Client Details
Trust Judgment in Applications

Problem:

In a vastly distributed infrastructure with vastly different properties, security depends not just on "who" but "how and where" authorized Alice asks to see a sensitive record.

Example:

What if authorized Alice asks to see a sensitive record from her mobile phone?

- from a public-access machine, that's physically hardened?

- from a trusted machine in the same locked room as the server?

- from a public-access machine in a college lounge?

- from a public-access machine in a college lounge in a college lounge?

(long-term relevance to ubiquitous computing)
Our approach: establish PKI for machines

Long-term: look at security-based transcoding

Attributes

demo server-side, app-side policy decisions based on security

Tool

Identify some interesting on-campus app area

Find policy language tool

Identify some interesting on-campus app area

Set up testbed to experiment with issuing security attribute certs for machines

Research Plan:

and easy for the application writer

and easy for the application writer

and easy for the application writer

and easy for the application writer

and easy for the application writer

In a way that is transparent to the user.

Infrastructure and transform accordingly

Our approach: establish PKI for machines

Application Details
Problem: PKI is about trust. Why should you?

Trusting Servers

Example: access to others' private material [Sma00]

Example: need-to-know databases [SmSa00]

Other end

A "trusted connection" doesn't help if you can't trust the server on the other end.

Example: how to efficiently provide material to authorized users...

Example: how to learn nothing, not even statistics, so that the server operator really is followed. even if insiders may have motivation to subvert it?

Consider not just advocacy groups, but also hostile expert witnesses...

What if the server operator may be motivated to attack?

What if the server operator would like to avoid liability?

What if an honest server operator would like to avoid liability?

(appplies throughout PKI base, and the apps built on it)
Our Approach: use high-end COTS secure coprocessors as trusted co-server, for

- use PKI to prove assurance
- move them to hardened devices

− keygen, key escrow, privacy data, proprietary algorithms...
− identity sensitive areas, with high insider attack risk...

Server Details

Research Plan:

• general-purpose computer at that level
• world’s first FIPS 140-1 Level 4—and (possibly) still the only
  prove it’s the real thing, doing the right thing
• guarded by sw/hw security arch that loads 3rd-party apps, and then
• general-purpose computing environment, with crypto support

Example: **WebALPS**—using coprocessor as trusted co-server, for

security and privacy in Web interactions

(Shan?)
Putting it all Together

Example:

- Sensitivemedicalresearchdata.migratesamongvariousWebALPS-hardenedservers.
  
  **User is connecting,** and sanitize content appropriately.
  
  - ...and the server needs to identify the security context of *how the server attributes*...not the server identity!
  - ...but the users need to verify the database identity and the hardened server attributes...not the server identity!

  This data...
  
  Authorized users (some of whom can delegate access) can view...

  - Migrates among various WebALPS-hardened servers.
  - Sensitive medical research data.

- We want to our research to result in real prototypes, solving real problems...
Chartered September 2000.

Looking for partners:

- Sources for real problems
- Sources for technology and equipment
- Sources for funding

Ramping up staffing, lab

Initial research work underway

---

Status