The Grey Project: Device-Enabled Authorization

Lujo Bauer
Research Scientist, CyLab
Carnegie Mellon University

Co-lead: Mike Reiter
Converged Mobile Devices ("Smartphones")

- **Converged mobile devices ("smartphones")**
  - Match wireless telephony to evolved OS or application environments
  - Include the ability to download data to local storage, run applications, and store user data beyond PIM capabilities
  - Leaders: Nokia, Motorola, Sony Ericsson, RIM, Samsung

- **Smartphones on a trajectory to “win” in the market**
  - IDC: Smartphones show “significant growth and future promise”, with a 75% increase in shipments from mid-2005 to mid-2006
  - Stand to inherit mobile phone market that shipped over 900 million units in 2006 [IDC]—or more than one phone per eight people in the world
The Real Digital Divide
Encourage the spread of mobile phones is the most sensible and effective response to the digital divide
The Economist, March 10, 2005

... The digital divide that really matters, then, is between those with access to a mobile network and those without. The good news is that the gap is closing fast. The UN has set a goal of 50% access by 2015, but a new report from the World Bank notes that 77% of the world’s population already lives within range of a mobile network.
New Applications on the Horizon

DoCoMo trials Sony Felica smart chip mobile phones

i4u, December 16, 2003

... One service being tested ... allows residents of a new apartment complex to use the FeliCa-equipped phones as keys to both the main entrance and their homes. The phone can also pay utility bills when swiped against a reader at the building's entrance.

Smart phones work like train tickets

AP, February 22, 2005

... With a service planned for launch in January next year, they'll be able to use their mobile phones in place of the cards to pay for their train fares ... Users will also be able to use their Suica-compatible cell phones to pay at some restaurants, convenience stores and shops. ... The service will later be expanded to include online shopping and reserved ticket purchases.
Our Take on This Vision: The Grey System

- Existing efforts utilize these devices as a replacement for existing mechanisms (charge card, physical keys, …)

- However, we believe this device-centric paradigm can support more flexible approaches than previously possible

- Goal: to use smartphones to intelligently control environment
  - Loan you my car without giving you my phone
  - Send money from my phone to my daughter’s phone
  - Give my secretary temporary access to my email without revealing information (e.g., password) that could be used at a later time
  - Use my phone to open my hotel room door, without ever stopping by the front desk

  …and do it all from a distance
Ongoing Deployment at CMU
Capabilities of Smartphones

- **Good data connectivity**
  - Bluetooth: short range radio link
  - 2.5G / 3G for wireless access to data networks
  - Messaging protocols: MMS and SMS

- **Growing computation power: public-key crypto is within range**
  - On a Nokia N70 using 1024-bit RSA
    - Signature: 1332ms
    - Verification: 40ms

- **Rich graphical input and output**
  - Graphical display and camera
Some Challenges

- **A sufficiently flexible authorization infrastructure**
  - Must support usual modes of access and delegation for each protection mechanism it is to replace, and more

- **Device theft**
  - Should ensure that stolen devices cannot be misused

- **Usability**
  - Human-to-device authentication
  - Device-to-device authentication
  - Access-control policy creation
Distributed Access Control

Demonstrate that Mike authorizes access.

Jon

Mike

Kevin

Jason

Scott

Mike’s Office, D208
Distributed Access Control

I allow Scott to enter.
-Mike

I want access.
-Scott

Demonstrate that Mike authorizes access.

Mike’s Office, D208
Distributed Access Control

Jon

I allow Scott to enter.
- Jason

Demonstrate that Mike authorizes access

I want access.
- Scott

Mike’s Office, D208

I delegate to Jason.
- Mike

Jason

I authorize access.

Scott
Challenges in Distributed Access Control

- How to represent and reason about credentials
- How to efficiently identify and retrieve a set of credentials that will allow access

Making things difficult... 
- Potentially many different ways to derive authority
- Credentials are distributed among signers
- Credentials may be created in response to a request
- May not want to show your credentials to everyone
A Sample Access-Control Logic

- The logic is inhabited by
  - Terms that denote principals and strings
  - Formulas that are either “true” or “false”

- Terms:
  
  \[
  t ::= s \mid p \\
  p ::= \text{key}(s) \mid p.s
  \]

  where \( s \) ranges over strings and \( p \) over principals

- Formulas:
  
  \[
  \phi ::= s \text{ signed } \phi' \mid p \text{ says } \phi' \\
  \phi' ::= \text{action}(s) \mid p \text{ speaksfor } p \mid \text{delegate}(p, p, s)
  \]

  where \( s \) ranges over strings and \( p \) over principals
Inference rules

\[\text{pubkey signed } F\]

\[\text{key (pubkey) says } F\]  \hspace{1cm} \text{(says-I)}

\[F\]

\[\text{A says } F\]  \hspace{1cm} \text{(says-I2)}

\[A \text{ says } (A.S \text{ says } F)\]

\[A.S \text{ says } F\]  \hspace{1cm} \text{(says-LN)}
A Sample Logic (cont.)

A says \((B \text{ speaks for } A)\) \hspace{1cm} B says \(F\) \hfill (speaksfor-E)

\[ \hspace{1cm} \] \hspace{1cm} A says \(F\)

A says \((B \text{ speaks for } A.S)\) \hspace{1cm} B says \(F\) \hfill (speaksfor-E2)

\[ \hspace{1cm} \] \hspace{1cm} A.S says \(F\)

A says delegates\((A, B, U)\) \hspace{1cm} B says open\((U)\) \hfill (delegate-E)

\[ \hspace{1cm} \] \hspace{1cm} A says open\((U)\)
Embedding in HOL (Example)

A says F ≡ ?

\[
\begin{align*}
&\text{pubkey signed fmla} \\
&\text{(name (pubkey)) says fmla} \\
&\text{A says F}
\end{align*}
\]

A says F       A says (F \rightarrow G)

\[
\begin{align*}
&\text{A says G}
\end{align*}
\]
Embedding in HOL (Example)

A says $F \equiv$

$$\forall S. \forall A' F' G K. (K \text{ signed } F' \rightarrow S (\text{name}(K),F)) \land (F' \rightarrow S (A',F')) \land ((S (A',F') \land S (A', (F' \rightarrow G))) \rightarrow S (A',G)) \rightarrow S (A,F)$$

$$F$$

$$\overline{A \text{ says } F \equiv \text{def2}_i (\forall_i \text{ says'} \ \forall_i A' F' G K...)}$$
A Sample Scenario

Scott is one of Mike’s Students

I need to grade the midterms for Mike’s class

Mike must authorize access

Mike

Scott

Prover

Prover

Prover

Checker

Mike’s Office, D208
Possible Methods of Authorization

1. Mike signed open(D208)

2. Mike signed (Scott speaks for Mike)

3. Mike signed delegate(Mike, Scott, D208)

4. Mike signed delegate(Mike, Mike.Students, D208)
   Mike signed (Scott speaks for Mike.Students)

5. Mike signed delegate(Mike, Lujo, D208)
   Lujo signed (…)

 zoning (D208)
Example Proof: Backward Chaining

Mike signed (Scott speaks for Mike.Students)

Scott signed open(D208)

Mike signed delegate(Mike, Mike.Students, D208)

Mike.Students says open(D208)

Mike says open(D208)
Traditional Approach to Proof Generation

- Previous approaches generate proof in a centralized manner

1. Hi, Please open D208
2. Prove Mike says open(D208)
3. Mike signed Scott speaksfor Mike?
   No
4. Mike signed delegate(Mike,Scott,D208)?
   No
5. Mike signed Scott speaksfor Mike.Secretary?
   No
6. Mike signed Lujo speaksfor Mike?
   ...

We classify this approach as “Eager”
Eager Is Inefficient

- Prover must request certificates without knowledge of which certificates are available or will be signed
  - Prover may request certificates for too much authority, or too little
  - Many requests may be necessary to determine proper credential

- Eager prover must guess until correct credential found
Insight #1: Be Lazy

[Bauer, Garriss, Reiter]

Eager:

“I’ll try to figure out how to open Mike’s door!”

Lazy:

“It’s Mike’s door. He should know how to open it. Let him figure it out…”
A Lazy Approach

- Ask Mike to prove *Mike says open*(D208)

- Mike’s prover may use its local knowledge to complete proof
  - Certificate lookups may now be performed locally, rather than remotely

- More generally, when reasoning about the beliefs of principal *A*,
  - Always ask *A* to prove *A says F*
Lazy Proof Generation

1. Hi, Please open D208
2. Prove Mike says open(D208)
3. Prove Scott says open(D208) → Mike says open(D208)
4. Proof of Scott says open(D208) → Mike says open(D208)
5. Proof of Mike says open(D208)
A Good Step, but Not Enough

- We initially attempted to deploy a lazy prover in our testbed and learned, quickly, that it was still not good enough.

Problems:

- Sometimes Alice should not be asked to prove Alice says $F$, since someone else might be more appropriate to ask.
  - Usually the human user better knows who to ask.

- If proof outsourcing is done inline, then it induces too much network traffic and interrupts too many users.

- Performance very sensitive to the tactics chosen.
  - Tactics are fragile to changes in policies.
Insight #2: Defer “Expensive” Proof Steps

- A subgoal is “expensive” to obtain if it involves
  - User interaction, or
  - Network communication to solicit a remote proof

- Before pursuing an “expensive” subgoal, make sure it will be useful toward the end goal

- Algorithmically, this is done by “assuming” the subgoal and proceeding with the rest of the proof
  - If the rest of the proof cannot be completed, then we have saved the expense of pursuing it
Proving with Markers

Mike signed (Scott speaks for Mike.Students)

Scott signed open(D208)

Mike signed delegate(Mike, Mike.Students, D208)

Mike.Students says open(D208)

Mike says open(D208)
Proving with Markers

Demonstrate that Mike authorizes access

Jon

Mike

Kevin

Jason

Scott

Mike’s Office, D208
Insight #3: Move Work Off the Critical Path

- **Pre-compute from locally known credentials:**
  - All true formulas
  - Trust relationships (e.g., Bob says open → Alice says open)

- **Allows us to constrain search space of Backward Chaining without sacrificing completeness**

- **Precomputing all true formulas**
  - Use variant called incremental forward chaining
  - Runs on a single credential at a time, adding newly inferred facts to the knowledge base
Precomputing Trust Relationships

- **Used to model the flow of authority, e.g.,**
  - If Alice says Bob speaks for Alice, then
    - Bob says F $\rightarrow$ Alice says F
  - If Alice says delegate(Alice, Bob, office)
    - Bob says open(office) $\rightarrow$ Alice says open(office)

- **Refer to these as paths**

- **A sequence of paths can be compressed**
  - If
    - Bob says F $\rightarrow$ Alice says F,
    - Alice says open(office) $\rightarrow$ Dept says open(office)
  - Then
    - Bob says open(office) $\rightarrow$ Dept says open(office)
Path Compression Example

The credential Alice signed delegate(Alice, Bob, door1) Becomes the path:

Bob says open(door1) Alice says open(door1)

ADD

Step 1: Add to tail

Bob says open(door1) Charlie says open(door1)

ADD

Charlie says open(door1) Charlie ...

Dave ...

Charlie ...

::
Path Compression Example

Step 2: Add to head

(From Step 1)

Bob says open(door1) Alice says open(door1)

Bob says open(door1) Charlie says open(door1)

Eve says open(door1) Charlie says open(door1)

ADD
Path Compression Example

Step 2: Add to head

(From Step 1)

Bob says open(door1) Alice says open(door1)
Bob says open(door1) Charlie says open(door1)
Eve says open(door1) Alice says open(door1)
ADD
Generalizing Proofs

- Caching only useful when trying to prove the exact goal again
- However, proofs of two similar goals may have the same shape
- Given a proof, can we generalize it to prove similar goals?
- I.e. given a proof of Mike says open(D208), can we use this proof to efficiently prove Mike says open(D210)?

- Yes, and we call it Automatic Tactic Generation
  - Simple form of modeling a user’s interaction with the system
Insight #4: Automatic Tactic Generation

Mike signed (Scott speaks for Mike.Students)

Scott signed open(D208)

Mike signed delegate(Mike, Mike.Students, D208)

Mike.Students says open(D208)

3. Mike signed delegate(Mike, Mike.Students, D208)

Mike says open(D208)
Analytical Results

- **Lazy offers substantially better performance despite theoretically equivalent proving ability**
  - When using the same tactics, the Lazy strategy will find a proof if the Eager strategy finds a proof

- **Lazy can continue to be useful in a heterogeneous scenario**
  - If a collection of provers with different tactics cooperate using the Lazy strategy, the set of provable goals does not decrease
Quantitative Analysis

- Want to measure the number of network interactions (requests) made by prover
  - Lazy Requests → Proof requests
  - Eager Requests → Credential lookups
- Simulate using a deployable policy
- Results averaged across all possible accesses
An Abstract Policy

- Based on real policy for ECE Department
- Delegations are made to roles, then principals bound to those roles
- CA run by university, binds principal names to public keys
- Simulate various policy sizes by specifying the branching coefficients for the tree rooted at CMU
- Numbers shown for balanced trees – unbalanced trees vary by less than 4% on average

\[(X,Y,Z) \text{ tree } = X \text{ dept heads, each with } Y \text{ floor managers, each with } Z \text{ ordinary users}\]
\[(3 + X + X \times Y + X \times Y \times Z \text{ total principals})\]
Eager vs Lazy: Initial Access

- Lazy makes half as many requests as Eager.
- Optimization: Cache completed proofs (Positive Cache).
- Most redundant requests fail.
- Optimization: Cache the fact that a request failed (Negative Cache).
- Negative Cache reduces queries by 75%.
Principals that Cache Results

Eager

Lazy
Access by a 2\textsuperscript{nd} Principal

![Diagram showing access by a 2\textsuperscript{nd} principal with different access levels and their corresponding average number of requests per access.]
Automatic Tactic Generation

- Sequential access of four resources by same principal
- (2,4,10) tree
- ATG produces proofs with a minimum # of requests
Credential Reuse

Mike: Scott may take one can of soda
**Problem: Credentials Can Be Reused**

- **Some credentials represent non-reusable authority**
  - $20
  - The right to board an airplane
  - The right to sign up for a class

- **Traditional notion of digitally signed certificates doesn’t suffice**

- **Challenges in implementing non-reusable credentials**
  - Enforcing non-reusability
    - Exactly *when* is a credential consumed?
  - Develop an enforcement mechanism
  - Representing non-reusable credentials in logic
“Consumable” Credentials [Bauer, Bowers, Garg, Pfenning, Reiter]

Mike: Scott may take one can of soda
Consumable Credentials

Ratifier

[Diagram showing a person selecting a consumable credential from a vending machine]
Consumable Credentials

Ratifier

Man with phone and credentials

Coca-Cola vending machine with prohibition symbol
Consumable Credentials

- **Enable bounded-use access-control policies**
  - Allow door to be opened N times
  - Allow students one soda/day free of charge
  - Allow payment if funds available

- **Expressed in policies using linear logic**

- **Implemented via ratification of proofs**
  - Ratification enforces “consumption”
  - Ratified credentials cannot be separated and used in other proofs
  - Atomic to prevent “wasteful” consumption
  - Borrows ideas from work on contract signing
Atomic Ratification
Conclusion

- Converged mobile devices “everywhere” are a safe bet
  - We need to figure out how to use them

- We are currently developing a system to deploy on the CMU campus to demonstrate ubiquitous access control using them
  - Observe, develop, and experiment with realistic access-control policies
  - Gather and leverage data on patterns of use and interaction
  - Evaluate and enhance usability
Conclusion (continued)

Thus far, the project embodies several advances

1. New “lazy” distributed proving methods in the context of a flexible authorization system
   - Can generate any proof that a centralized approach can
   - Far more efficient than prior work, particularly when paired with automatic tactic generation

2. “Consumable” credentials to represent non-reusable authority
   - Ratifiers verify that credentials are valid before a proof is complete
Status

- Deployment of Grey is currently in progress on second floor of CMU’s Collaborative Innovation Center
  - Approximately 35 Grey-capable doors and 20+ users at the moment

- Also have developed Grey-compatible Windows XP and Linux login modules
  - Access-control module for web servers under development
Thanks for your attention!

Questions?