PURDUE UNIVERSITY

CS52600: Information Security

Authentication and Identity

4 October, 2010

Prof. Chris Clifton

What we have so far: Access control

- Given subjects *S*, objects *O*
  - What subject is allowed
  - what type of access
  - to what object
- Enough for a realistic system?
  - How do we know a request really is from *s*?
  - And the object accessed really is *o*?
- Next Topic: Authentication and Identity

2

What is Authentication?

- Authentication: Binding of identity to subject
- How do we do it?
  - Entity *knows* something
    - Passwords, id numbers
  - Entity *has* something
    - Badge, smart card
  - Entity *is* something
    - Biometrics
  - Entity is *someplace*
    - Source IP, restricted area terminal

3

Authentication System: Formal Definition

- *A*: set of *authentication information*
  - used by entities
- *C*: set of *complementary information*
  - used by system to validate authentication information
- *F*: Set of *complementation functions*
  - $f : A \rightarrow C$
  - Generate appropriate $c \in C$ given $a \in A$
- *L*: set of *authentication functions*
  - $l: A \times C \rightarrow \{$ **true, false** $\}$
  - verify identity
- *S*: set of *selection functions*
  - $s : ? \rightarrow A$
  - Generate/alter *A* and *C*

4

Authentication System

- $(A, C, F, L, S, (f, l))$ such that
  - $\forall f \in F, \forall l \in L, \exists (f, l)$ in the system such that $\forall a \in A, \forall c \neq f(a) \in C$:
    - $l(a, f(a)) \rightarrow$ **true**
    - $l(a, c) \rightarrow$ **false**
    - with high probability
- (Bad) example: plaintext passwords
  - $A = C$ = alphabet*
  - f returns argument
  - l is string equivalence

5

Background: Cryptography

- Encryption system $E_k(s)$
  - over all choices of k, should form uniform distribution
  - Thus "appears" independent of s
- Decryption: $D_k(E_k(s)) = s$
- Strength: Given $E_k(s)$, finding k or s difficult
  - Better: given s, E, $E_k(s)$, learning k hard
  - choosing k, computing D_k, E_k must be easy
- Public key: p, r such that $D_r(E_p(s)) = s$
  - Similar notions of strength

6

Authentication Systems: Passwords

- Information known to subject
- Complementation Function
  - Identity – requires that *c* be protected
  - One-way hash – function such that
    - $f(a)$ easy to compute
    - $f^{-1}(c)$ difficult to compute
  - Better ideas?

7

Attacks on Passwords

- Dictionary attack: Trial and error guessing
  - Type 1: attacker knows *A*, *f*, *c*
  - Type 2: attacker knows *A*, *I*
  - Difficulty based on $|A|$, time *g* to compute *f* or *I*
    - Probability *P* of breaking in time *T*: $P \geq Tg/|A|$
    - Problem: often smaller in practice than in theory
- Password Selection
  - Random
  - Pronounceable nonsense
  - User selection
    - Controls on allowable
  - Password checking, aging

8

Authentication Systems: Challenge-Response

- Pass algorithm
  - authenticator sends message *m*
  - subject responds with *a*(*m*)
    - *a* is an encryption function
    - In practice: key known only to subject
- What is the advantage over passwords?
  - Avoids "replay" attacks
- One-time password
  - *a* changes after use
  - Why is this challenge-response?

9

Attacks on Challenge-Response

- Type 1 attack
  - Attacker knows (space of) encryption function
  - Captures challenge and response
  - Learns encryption function / key
  - *Can now properly respond to new challenge*
- Solution: encrypt challenge
  - Use shared key to share session key
  - Session key encrypts challenge
  - Challenge thus indistinguishable from random data

10

Authentication Systems: Biometrics

- Used for human subject identification
- Based on physical characteristics that are tough to copy
  - fingerprint
  - voice patterns
  - iris/retina patterns
  - face recognition
  - keystroke interval/timing/pressure

11

Attacks on Biometrics

- Fake biometrics
  - fingerprint “mask”
  - copy keystroke pattern
- Fake the interaction between device and system
  - Replay attack
    - *Authenticate the authenticator!*
  - Requires careful design of entire authentication system

12

Authentication Systems: Location

- Based on knowing physical location of subject
- Example: Secured area
  - Assumes separate authentication for subject to enter area
  - In practice: early implementation of challenge/response and biometrics
- What about generalizing this?
  - Assume subject allowed access from limited geographic area
    - I can work from (near) home
  - Issue GPS Smart-Card
  - Authentication tests if smart-card generated signature within spatio/temporal constraints
  - Key: authorized locations known/approved in advance
    - *If I know I'll be traveling, login from my office disabled!*

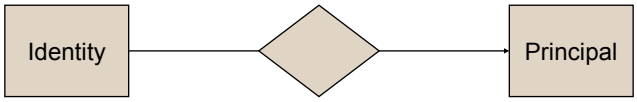
13

Authentication vs. Identity

- Authentication: Binding of identity to subject
  - We know how
  - We've discussed subjects
  - *But what precisely is identity?*
- Principal: Unique Entity
  - Subject
  - Object
- Identity: Specifies a principal

14

Identity = Principal?



- Identity to Principal may be many-to-one
  - Given identity, know principal
  - Other direction unimportant?
- Examples: Unix
  - User identity
  - File identity

15

What is a Principal?

- Entity
  - Subject
  - Object
- Also a set of entities
  - Think of subject as “power user”, not individual
- Examples:
  - Groups: defined collection of users with common privileges
  - Roles: membership tied to function

16

Representing Identity

- Randomly chosen: not useful to humans
- User-chosen: probably not unique
  - At least globally
- Hierarchical: Disambiguate based on levels
  - File systems
  - X.500: Distinguished Names
    - /O=Purdue University/OU=Computer Sciences/CN=clifton
  - Aliases
    - /O=Purdue University/OU=ITaP/CN=cclifton

17

Validating Identity

- Authentication: Does subject match purported identity?
- Problem: Does identity match principal?
- Solution: *certificates*
  - Certificate: Identity validated to belong to known principal
  - Certification Authority: Certificate Issuer
    - Authentication Policy: describes authentication required to ensure principal correct
    - Issuance policy: Who certificates will be issued to
  - CA is *trusted*

19

Certificate Implementation

- Is a certificate real?
  - Digital signatures
  - Certificate = Identity + $E_{IssuerPrivateKey}$(Identity)
    - Correct if Identity = $D_{IssuerPublicKey}$(Signature)
- Can I trust it?
  - Hierarchy of issuers
    - Certificate includes certificate of issuer chain
  - Higher levels place (contractual) conditions on lower level issuance
    - Common issuance, authentication policy
  - *Also solves uniqueness issue*

20

Certificate Examples

- Verisign
  - Independently verifies identity of principal
  - Levels of certification
    - Email address verified
    - Name/address verified
    - Legal identity verified
  - More common: *corporate* identity
    - Is this really PayTuition.EDU I'm giving my bank account number to?
- PGP (Pretty Good Privacy): "Web of Trust"
  - Users verify/sign certificates of other users
  - Do I trust the signer?
    - *Or someone who signed their certificate?*

21

Internet Identity

- Host Identity: Who is this on the network?
- Ethernet: MAC address
  - Guarantees uniqueness
- IP address: aaa.bbb.ccc.ddd
  - Provides hierarchy to ease location
- Domain Name Service
  - Human-recognizable name to IP
  - aliasing
- Issues: Spoofing
  - At any level, change mapping of identity to principal

22

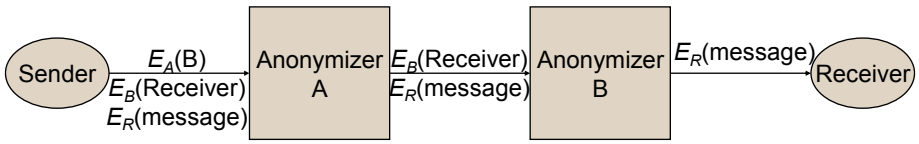
Anonymity

- What if identity not needed?
  - Web browsing
  - Complaints about assignments
- Removing identity not as easy as it sounds
  - I can send email without my userid
  - But it still traces back to my machine
- Solution: anonymizer
  - Strips identity from message
  - Replaces with (generated) id
  - Send to original destination
  - Response: map generated id back to original identity

23

Anonymity

- Problem: Anonymizer knows identity
  - Can it be trusted?
  - *Courts say no!*
- Solution: multiple anonymizers
  - Onion Routing
  - Crowds



24