## Summary From the Last Lecture

- Key exchange
  - Kerberos
  - Digital certificates
- Certificate authority structure
  - PGP, hierarchical model
- Recovery from exposed keys
  - Revocation lists, time-limited keys, real time validation
- Group key management
  - Robustness, forward/backward secrecy

## Authentication

## Basis for Authentication

- Ideally
  - Who you are
- Practically
  - Something you know
  - Something you have
  - Something about you

## Something You Know

- Password or Algorithm
  - e.g. encryption key derived from password
- Issues
  - Someone else may learn it
    - Find it, sniff it, trick you into providing it
  - Other party must know how to check
  - You must remember it

## Password Authentication

- Alice inputs her password, computer verifies this against list of passwords
- If computer is broken into, hackers can learn everybody's passwords
  - Use one-way functions, store the result for every valid password
  - Perform one-way function on input, compare result against the list

## Password Authentication

- Hackers can compile a list of frequently used passwords, apply one-way function to each and store them in a table – dictionary attack
- Host adds random salt to password, applies one-way function to that and stores result and salt value
  - Randomly generated, unique and long enough

## Password Authentication

- Someone sniffing on the network can learn the password
- Lamport hash or S-KEY – time-varying pass
  - To set-up the system, Alice enters random number *R*
  - Host calculates $x_0=h(R)$, $x_1=h(h(R))$, $x_2=h(h(h(R)))$,…, $x_{100}$
  - Alice keeps this list, host sets her password to $x_{101}$
  - Alice logs on with $x_{100}$, host verifies $h(x_{100})=x_{101}$, resets password to $x_{100}$
  - Next time Alice logs on with $x_{99}$

## Password Authentication

- Someone sniffing on the network can learn the password
  - Host keeps a file of every user's public key
  - Users keep their private keys
  - When Alice attempts to log on, host sends her a random number *R*
  - Alice encrypts *R* with her private key and sends to host
  - Host can now verify her identity by decrypting the message and retrieving *R*

## Public Key Authentication

- Key Distribution
  - Confidentiality not needed for public key
  - Solves $n^2$ problem
- Performance
  - Slower than conventional cryptography
  - Implementations used for key distribution, than use conventional crypto for data encryption
- Trusted third party still needed
  - To certify public key
  - To manage revocation
  - In some cases, third party may be off-line

## Single Sign-On

- Passport
- Liberty Alliance
- Shibboleth

## Federated Identity Passport vs Liberty Alliance

- Two versions of Passport
  - Centralized and federated
- Liberty Alliance
  - Loosely federated with framework to describe authentication provided by others

## Passport v1

- Goal is single sign-on
  - Solves problem of weak or repeated user/pass combinations
- Implemented via redirections
  - Users authenticate themselves to a common server, which gives them tickets
  - Similar flavor to Kerberos but different environment – many organizations
- Widely deployed by Microsoft
  - Designed to use existing technologies in servers/browsers (HTTP redirect, SSL, cookies, Javascript)

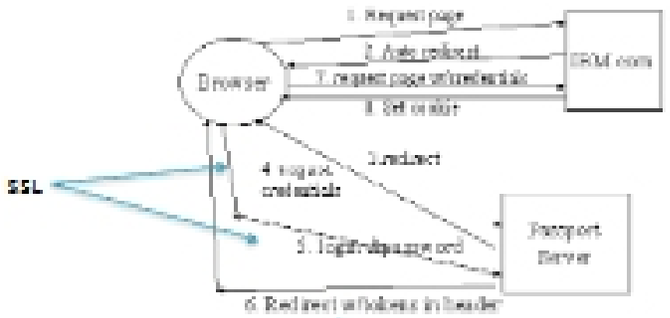
David P. Kormann and Aviel D. Rubin, Risks of the Passport Single Signon Protocol, Computer Networks, Elsevier Science Press, volume 33, pages 51–58, 2000.

## How Passport Works

- Client (browser), merchant (Web server), Passport login server
- Passport server maintains authentication info for client
  - Gives merchant access when permitted by client
- Divides client data into profile (address) and wallet (credit card)

David P. Kormann and Aviel D. Rubin, Risks of the Passport Single Signon Protocol, Computer Networks, Elsevier Science Press, volume 33, pages 51–58, 2000.

## How Passport Works




David P. Kormann and Aviel D. Rubin, Risks of the Passport Single Signon Protocol, Computer Networks, Elsevier Science Press, volume 33, pages 51–58, 2000.

## Some Problems with Passport

- User interface is confusing and may misrepresent the reality
- Weak keys may be used for 3DES
- Single key is used to encrypt cookies for all clients
- Cookies stay on machine, can be stolen
  - No authenticator (timestamp), like in Kerberos, enables reuse by others
- Coupling of Hotmail with Passport

Read more at http://avirubin.com/passport.html

## Federated Passport

- Multiple federated identity providers
  - E.g. ISPs register own users
  - One can rely on claims made by other ID providers
- Claims
  - Emails, relationships, authorization for scenarios, ownership of private/public key pair
- Need "translators" for different claim languages

## Liberty Alliance

- Design criteria was most of the issues addressed by Federated Passport, i.e. no central authority
- Use SAML (Security Association Markup Language) to describe trust across authorities, and what assertions mean from particular authorities
- Four assurance levels
  - How much we trust a given identity assertion
  - Little, some, high and very high confidence