17-500 Usable Privacy and Security
Authentication overview
Presented on March 28, 2006 by **Dr. Michael Reiter**
Scribe Notes by: Ricardo Villamarin-Salomon

(Note: these scribe notes try to cover mainly the questions/discussions during class; it does not give too many details of the lecture notes themselves, because the slides are self-explanatory or have explicit references. Only some complementary information has been added mainly in the first sections)

Toward Fixing the Compliance Defects of Public Key Cryptography

Compliance Defects in PKI

A compliance defect in a cryptosystem is a rule of operation that is both difficult to follow and unenforceable.

According to Don Davis[1], Public Key cryptography has five unrealistic rules of use that he calls compliance defects that correspond one for one with the crucial moments in a key-pair's life-cycle:

1. Authenticating the User (Issuance): How does a CA authenticate a distant user, when issuing an initial certificate?
2. Authenticating the CA (Validation): Public-key cryptography cannot secure the distribution and validation of the Root CA's public key.
3. Certificate Revocation Lists (Revocation): Timely and secure revocation presents terrific scaling and performance problems. As a result public-key deployment is proceeding with out a revocation infrastructure
4. Private-Key Management (Single-Sign On): The user must keep his long-lived private key in memory throughout his login-session
5. Passphrase Quality (PW-Change): There's no way to force a public ┌ key user to choose a good passphrase

It is possible to fix a compliance defect by:

- Improve the user interface
- Impose an enforcement mechanism

For this last point and the problem of verifying the root's public key, this is difficult given the scale of the Internet. A solution initially by Netscape (NS) browsers is to store the public key of several CAs

[1] Davis D. (1996). "Compliance Defects in Public-Key Cryptography" Proc. 6th Usenix Security Symp., San Jose CA, 1996

> As an anecdote, Dr. Reiter referred the case where NS incorporated this information from AT&T in NS3/4 and that Microsoft just copied that information without knowing more details about it.

Unfortunately also, some changes in the UI have been for the worse not the better.

Improving the User Interface

Snowflakes:
Used to display hash outputs. Each hash should correspond to a unique snowflake. If any two snowflakes can be distinguished on careful inspection, then the cryptographic soundness is equivalent to that of the hash.
The design is based on the principle of making it as easy as possible to distinguish any two snowflakes. The most efficient method, from an information theoretic point of view, would be to just lay the bits down on a grid. However, this would just look like random bits to most people. So it has two additional constraints. First, the snowflake has D6 symmetry (the same as real snowflakes). The second constraint is that the snowflake is connected - no isolated black triangles. This encourages the bits to fall into patterns that may be more recognizable to the eye.

Question raised: Are snowflakes easier than keys?
Problem: No user studies:

http://ftp.cerias.purdue.edu/pub/lists/best-of-security/47

Random Art:
Given an initial seed, Random Art generates a random mathematical formula which defines the color value for each pixel on the image plane. The image generation process is deterministic and the image depends only on the initial seed. It is therefore not necessary to store the images pixel-by-pixel, since the image can be computed quickly from the seed. All images are hand-selected to ensure consistent quality.

> A user study showed that although it took more time to users to create Random-Art based passwords than text passwords and PINs, and that it also took longer for users to authenticate with them, after one week of using passwords/PINs and Random Art for authentication, there was a greater degradation in performance with PINs and passwords compared to random arts[2].

Matt DeSantis: "So, does the grammar help out in creating pixels that are some factor similar to the one before? Because if it doesn't then the program would create ‘noise’ each time and that would be difficult for a user to distinguish”.

Graphical Passwords:
“A memorable password is one for which there exists a short algorithm to generate it.”

---

[2] R. Dhamija and A. Perrig. Deja Vu: A User Study Using Images for Authentication. In 9th USENIX Security Symposium, 2000.

Question: “Why to memorize the passwords?”, “why not storing them?”
Sasha Romanosky: Smart cards?
Mike Reiter: Smart cards as “password management systems”.

Mike Reiter: “how can we tell if users are more successful at using graphical passwords vs. string passwords?”

Matt DeSantis: How would you know unless you asked people to self-report. Meaning - If all he cares about is a view into whether a high-entropy password was encoded with whatever method, why does it matter, or why do you care?

Mike Reiter: People tend to draw symmetric pictures.

Making the Old Interface More Effective

Two approaches that were discussed
- Use the network
- Use the user

Use the network
“Store private key in a protected server that authenticates user before sending the private key”

Mike Reiter: What is the bad thing about this? (slide #18)

Kami Vaniea: Easy to do a denial of service attack. The account of a user may lock up after more than three unsuccessful tries

Reducing Trust in the Server

Keep the key at the client, but in a disabled state

Mike Reiter: how can this be implemented?

Kami Vaniea: Using private keys

Reducing Trust in the Client

Sasha Romanosky: Whether or not online revocation lists would solve the problem of a malicious insider signing and decrypting. Mike Reiter confirmed that while this was part of the solution, the revocation lists are only useful if people request and abide by them.