

Lecture 11

Authentication

1

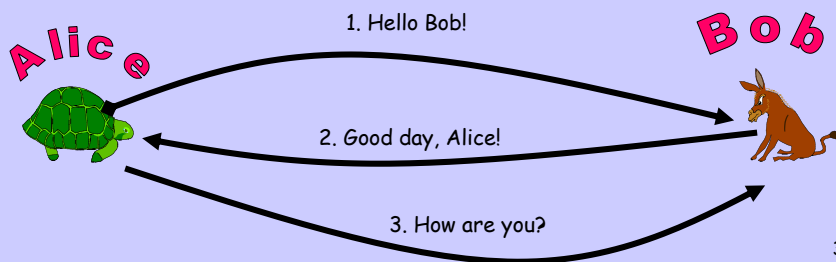
Where are we now?

- We “know” a bit of the following:
 - Conventional cryptography
 - Hash functions and MACs
 - Public key cryptography
 - Encryption
 - Signatures
 - Identification (Fiat-Shamir) + Zero Knowledge
- And now what?
 - Protocols
 - Authentication/Identification
 - Key distribution

2

Secure Protocols

- A **protocol** is a set of rules for exchanging messages between 2 or more **entities**
- A protocol has a number of **rounds** (>1) and a number of **messages** (>1)



Secure Protocols

- A **message** is a unit of information send from one entity to another as part of a protocol
- A **round** is a basic unit of protocol time:
 1. *Wake up because of:*
 - a) *Alarm clock*
 - b) *Initial start or*
 - c) *Receive message(s) from other(s)*
 2. *Compute something*
 3. *Send message(s) to others*
 4. *Repeat steps 2-3, if needed*
 5. *Wait for message(s) or sleep until alarm clock*

What's a *secure* protocol?

- When acting honestly, *entities* (participants) achieve the stated **goal** of the protocol, e.g.:
 - A successfully authenticates to B,
 - A and B exchange a fresh session key
- Adversary can defeat this goal
 - e.g., by successfully impersonating A in an authentication protocol with B

5

The Entities (2-party setting)

- **Alice** and **Bob**
 - want to mutually authenticate and/or share a key
- **Eve**, the adversary
 - passive or active
- In more complex protocols, **TTP**
 - 3rd party trusted by both Alice and Bob

6

Definitions

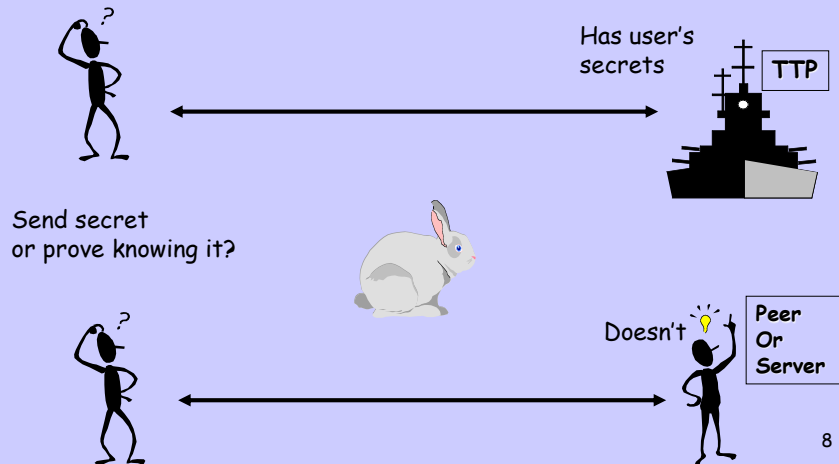
- **Entity authentication:**
 - corroboration that an entity is the one claimed.
- **Unilateral authentication:**
 - entity authentication: providing one entity with assurance of the other's identity, but not vice versa
- **Mutual authentication:**
 - entity authentication which provides both entities with assurance of each other's identity

7

Purpose

Examples:

- Bank transactions, e.g., cash withdrawals
- Remote login
- File access
- P2P transaction



8

Basis for Authentication

- Something you **know** (a PIN, or password).
- Something you **have**:
 - A secure token, e.g., that generates a one-time password.
 - key embedded in a "secure area" on a computer, in browser software, etc.
 - a smartcard (which may contain keys and can perform cryptographic operations on behalf of a user).
- Something you **are** (a biometric).

9

Concrete Scenarios

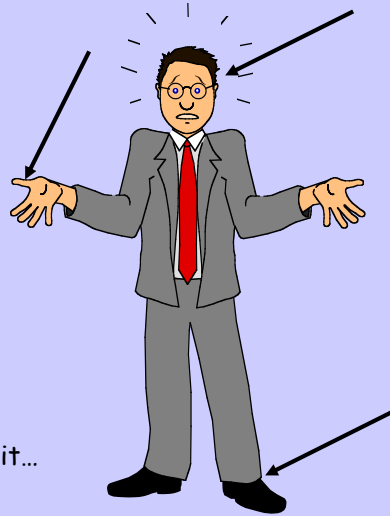
- ❖ PIN-, PW-, Biometric-based schemes
 - ❖ Kerberos (covered later)
 - ❖ SecureID tokens
 - ❖ Iris/retina scanners
 - ❖ Thumbprint & Handprint
 - ❖ Handwriting acceleration & pressure
- ❖ Public Key Identification Schemes:
 - ❖ Fiat-Shamir, etc.
- ❖ Authentication protocols
 - ❖ conventional- and public key-based (covered later)

Human Failings

- ❖ Humans are notoriously unreliable
- ❖ Human memory is very volatile storage

What a human can remember:

- ❖ PIN (no more than 6-8 digits)
- ❖ Password (a word or a short phrase)
- ❖ Can a human do single-digit sums? Forget it...



11

Biometrics

- Accuracy:
 - False acceptance rate.
 - False rejection rate.
- Retinal scanner, fingerprint reader, handprint reader, voiceprint, keystroke timing, signature (shape or pressure), etc.

12

Fingerprints

- Vulnerability:
 - Dummy fingers and dead fingers
- Suitability and stability:
 - Not for people with high probability of damaged fingerprints (e.g., exema)
 - Not for kids who are still growing

13

Voice Recognition

- Single phrase:
 - Can use tape recorder to fake
- Stability:
 - Background noise
 - Colds, vocal cord damage/strain, laughing gas 😊
 - Use with public phones

14

Keystroke Timing

- Each person has a distinct typing timing and style
 - Hand/finger movements
- Suitability:
 - Best done for "local" authentication
 - Avoid network traffic delay

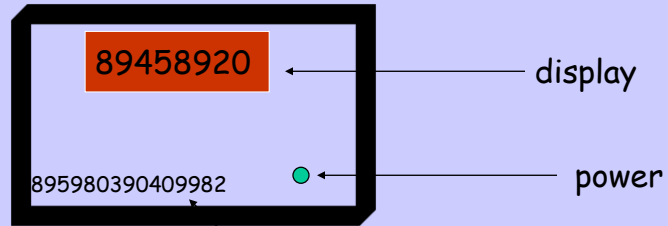
15

(non-digital) Signatures

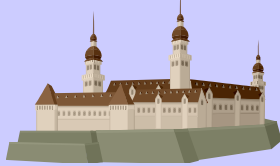
- Machines can't match human experts in recognizing shapes of signatures
- Add information on acceleration and/or pressure
 - Signing on a special electronic tablet

16

SecureID



TTP/Server:
secure & knows all secrets!



Serial #
Id-based key
(inside)