CS306: Introduction to IT Security

Fall 2018

Lecture 7: User authentication

Instructor: **Nikos Triandopoulos**

October 23, 2018

STEVENS
INSTITUTE *of* TECHNOLOGY
THE INNOVATION UNIVERSITY®

Last lecture

- Public-key cryptography
  - motivation
  - public-key encryption
    - hybrid encryption
    - ElGamal encryption scheme
  - digital signatures
  - public-key certificates
  - key agreement

Today

- Revision
  - midterm exam
- User authentication

7.0 Announcements

CS306: Tentative Syllabus

Week	Date	Topics	Reading	Assignment
1	Aug 28	Introduction	Ch. 1	-
2	Sep 4	Symmetric encryption	Ch. 2 & 12	Lab 1
3	Sep 11	Symmetric encryption II	Ch. 2 & 12	Lab 2, HW 1
4	Sep 18	Message authentication	Ch. 2 & 12	Lab 3, HW 1
5	Sep 25	Hash functions	Ch. 2 & 12	Lab 4
6	Oct 2	Public-key cryptography	Ch. 2 & 12	Lab 5
–	Oct 9	No class (Monday schedule)		Help session
7	Oct 16	Midterm (closed books)	All materials covered	No labs

CS306: Tentative Syllabus (continued)

Week	Date	Topics	Reading	Assignment
8	**Oct 23**	**Authentication**	**Ch. 2**	**No labs**
9	Oct 30	Access control		
10	Nov 6	Software, Web & Network security		
11	Nov 13	Database & cloud security		
12	Nov 20	Privacy		
13	Nov 27	Economics		
14	Dec 4	Legal & ethical issues		
15	Dec 11 (or later)	**Final** (closed books)	All materials covered*	

* w/ focus on what covered after midterm

CS306: Course outcomes

- **Terms**
  - describe common security terms and concepts
- **Cryptography**
  - state basics/fundamentals about secret and public key cryptography concepts
- **Attack & Defense**
  - acquire basic understanding for attack techniques and defense mechanisms
- **Impact**
  - acquire an understanding for the broader impact of security and its integral connection to other fields in computer science (such as software engineering, databases, operating systems) as well as other disciplines including STEM, economics, and law
- Ethics
  - acquire an understanding for ethical issues in cyber-security

Questions?

7.1 User authentication

User identification & authentication

Identification

- asserting who a person is

Authentication

- proving that a user is who she says she is
- methods
  - something the user *knows*
  - something the user *is*
  - something user *has*

Does authentication imply identification?

Suppose that a user

- provides her (login) name and
- uses one of the three methods to authenticate into a computer system
  - either terminal or remote server via a web browser
- when does user authentication imply user identification?

Issues with “Something you know”

- the user has to know some secret to be authenticated
  - password, personal identification number (PIN), personal information like home address, date of birth, name of spouse (“security” questions)
- anybody who obtains your secret “is you”
  - impersonation Vs. delegation
- you leave no trace if you pass your secret to somebody else
- what if there is a case of computer misuse?
  - i.e., where somebody has logged in using your username and password…
  - can you prove your innocence?
  - can you prove that you have not divulged your password?

Thus…

- a password does not authenticate a person
- successful authentication only implies that the user knew a particular secret
- there is no way of telling the difference between the legitimate user and an intruder who has obtained that user's password
- **unfortunately: this holds true for almost all of authentication methods…**

Attacks on “Something you know”

- passwords
  - or PINs
  - or answers to “security” questions (e.g., where did you meet your wife?)
- attacks
  - inferring likely passwords/answers, guessing
    - low-entropy secrets (e.g., password is “Password1”)
  - defeating concealment
    - leaked password files
  - using exhaustive or brute-force attack, rainbow tables, or dictionary attacks
    - impose different time-space trade-offs on attacker

Counteracting online dictionary attacks

- "online" brute-force or dictionary attack against passwords
  - employs only the authentication system
  - the attacker tries to impersonate a victim by trying
    - all possible (short length) passwords or
    - passwords coming from a known dictionary
- “offline” brute-force or dictionary attack
  - employs a leaked file of hashed passwords
- can online attacks be prevented and, if yes, how?

Phishing & spoofing

- identification and authentication through username and password provide **unilateral authentication**
- computer verifies the user's identity but the user has no guarantees about the identity of the party that has received the password
- in **phishing** and **spoofing** attacks a party voluntarily sends the password over a channel, but is misled about the end point of the channel

Spoofing

- attacker starts a malicious program that presents a fake login screen and leaves the computer
- if the next user coming to this machine enters username and password on the fake login screen, these values are captured by the malicious program
  - login is then typically aborted with a (fake) error message and the spoofing program terminates
  - control returns to operating system, which now prompts the user with a genuine login request
  - thus, the victim does not suspect that something wrong has happened
    - the victim may think that the password was mistyped…

Counteracting password spoofing

- display **number of failed logins**
  - may indicate to the user that an attack has happened
- **trusted path**
  - guarantee that user communicates with the operating system and not with a spoofing program
- **mutual authentication**
  - user authenticated to system, system authenticated to user

Phishing

- attacker impersonates the system to trick a user into releasing the password
- e.g.,
  - a message could claim to come from a service you are using
  - tell you about an upgrade of the security procedures
  - and ask you to enter your username and password
    at the new security site that will offer stronger protection
- attacker impersonates the user to trick a system operator into releasing the password to the attacker
  - **social engineering**

Password storage

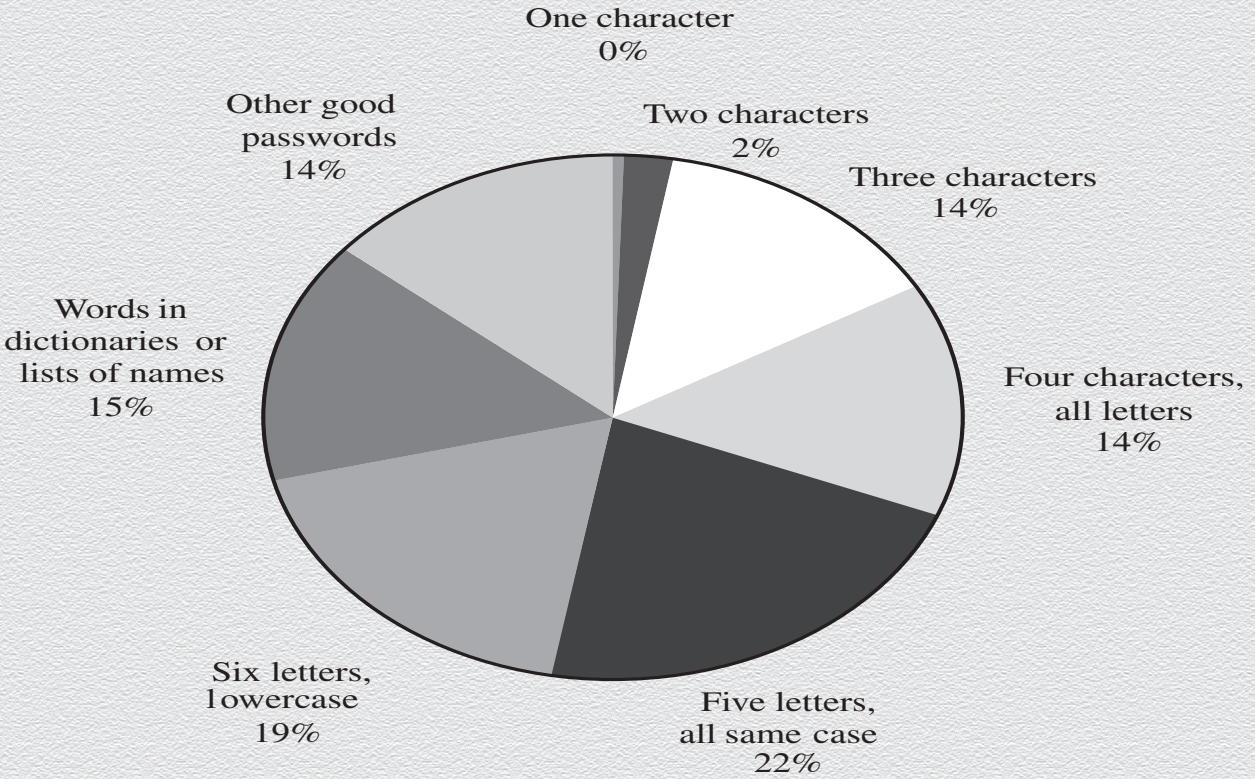
Identity	Password
Jane	qwerty
Pat	aaaaaa
Phillip	oct31witch
Roz	aaaaaa
Herman	guessme
Claire	aq3wm$oto!4

Plaintext

Identity	Password
Jane	0x471aa2d2
Pat	0x13b9c32f
Phillip	0x01c142be
Roz	0x13b9c32f
Herman	0x5202aae2
Claire	0x488b8c27

Concealed

Distribution of password types



Hashing passwords is not enough

An immediate control against password leakage through stolen password files, involves concealing passwords stored at the authentication server via hashing

Why are offline dictionary attacks quite effective using leaked hashed passwords in practice?

Countermeasures

Password salting

- to slow down dictionary attacks
  - a user-specific **salt** is appended to a user's password before it is being hashed
  - each salt value is stored in the clear along with its corresponding hashed password
  - if two users have the same password, they will have different hashed passwords
  - example: Unix uses a 12 bit salt

Hash strengthening

- to slow down dictionary attacks
  - a password is hashed k times before being stored

Protecting the password file

Operating system maintains a password file (with user names and passwords)

- attacker could try to compromise its confidentiality or integrity
- options for protecting the password file
  - cryptographic protection
  - access control enforced by the operating system
  - combination of cryptographic protection and access control, possibly with further measures to slow down dictionary attacks

Access control settings

- only privileged users must have write access to the password file
  - an attacker could get access to the data of other users simply by changing their password
  - even if it is protected by cryptographic means
- if read access is restricted to privileged users, passwords could be stored unencrypted
  - in theory – in practice, bad idea because of breaches
- if password file contains data required by unprivileged users, passwords must be "encrypted"; such a file can still be used in dictionary attacks
  - typical example is **/etc/passwd** in Unix
  - many Unix versions store encrypted passwords in a shadow password file (not publicly accessible)

Caching passwords

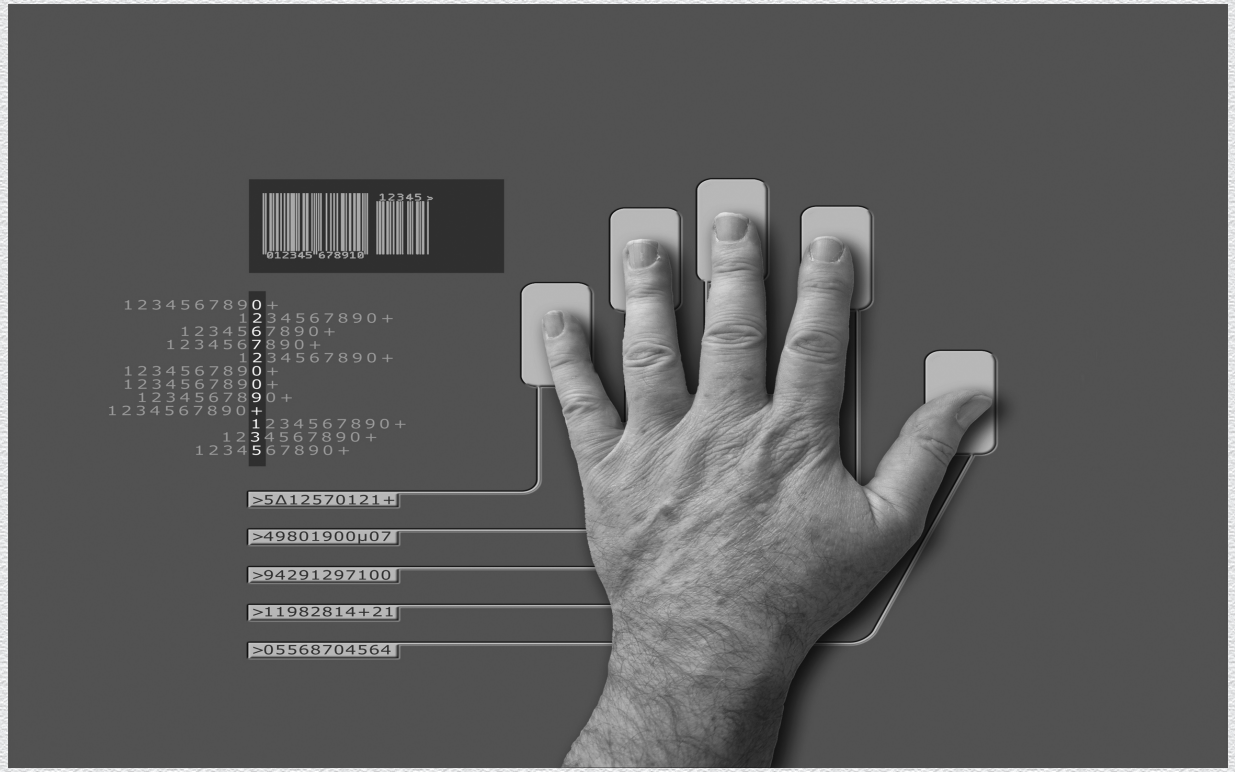
- description of login has been quite abstract
  - password travels directly from user to the password checking routine
- in reality, it will be held temporarily in intermediate storage locations
  - e.g., like buffers, caches, or a web page
- management of these storage locations is normally beyond user's control
  - a password may be kept longer than the user has bargained for

7.2 Something you are – biometric authentication

Something you are

- biometric schemes use people's unique physical characteristics
  - traits, features
  - face, finger prints, iris patterns, hand geometry
- biometrics may seem to be the most secure solution for user authentication
- biometric schemes are still quite new

Biometrics: Something you are



Problems with biometrics

- Intrusive
- Expensive
- Single point of failure
- Sampling error
- False readings
- Speed
- Forgery

Fingerprint

- Enrolment
  - reference sample of the user's fingerprint is acquired at a fingerprint reader
- Features are derived from the sample
  - fingerprint minutiae
    - end points of ridges, bifurcation points, core, delta, loops, whorls, …
- For higher accuracy, record features for more than one finger
- Feature vectors are stored in a secure database
- When the user logs on, a new reading of the fingerprint is taken
  - features are compared against the reference features

Identification Vs. verification

- Biometrics are used for two purposes
  - Identification: 1:n comparison, i.e., identify user from a database of n persons
  - Verification: 1:1 comparison, i.e., check whether there is a match for a given user
- Authentication by password
  - clear reject or accept at each authentication attempt
- Biometrics
  - stored reference features will hardly ever match precisely features derived from the current measurements

Failure rates

- Measure similarity between reference features and current features
- User is accepted if match is above a predefined threshold
- **New issue: false positives and false negatives**
- Accept wrong user (false positive)
  - security problem
- Reject legitimate user (false negative)
  - creates embarrassment and an inefficient work environment

Forgeries

Fingerprints, and biometric traits in general, may be unique but they are no secrets!

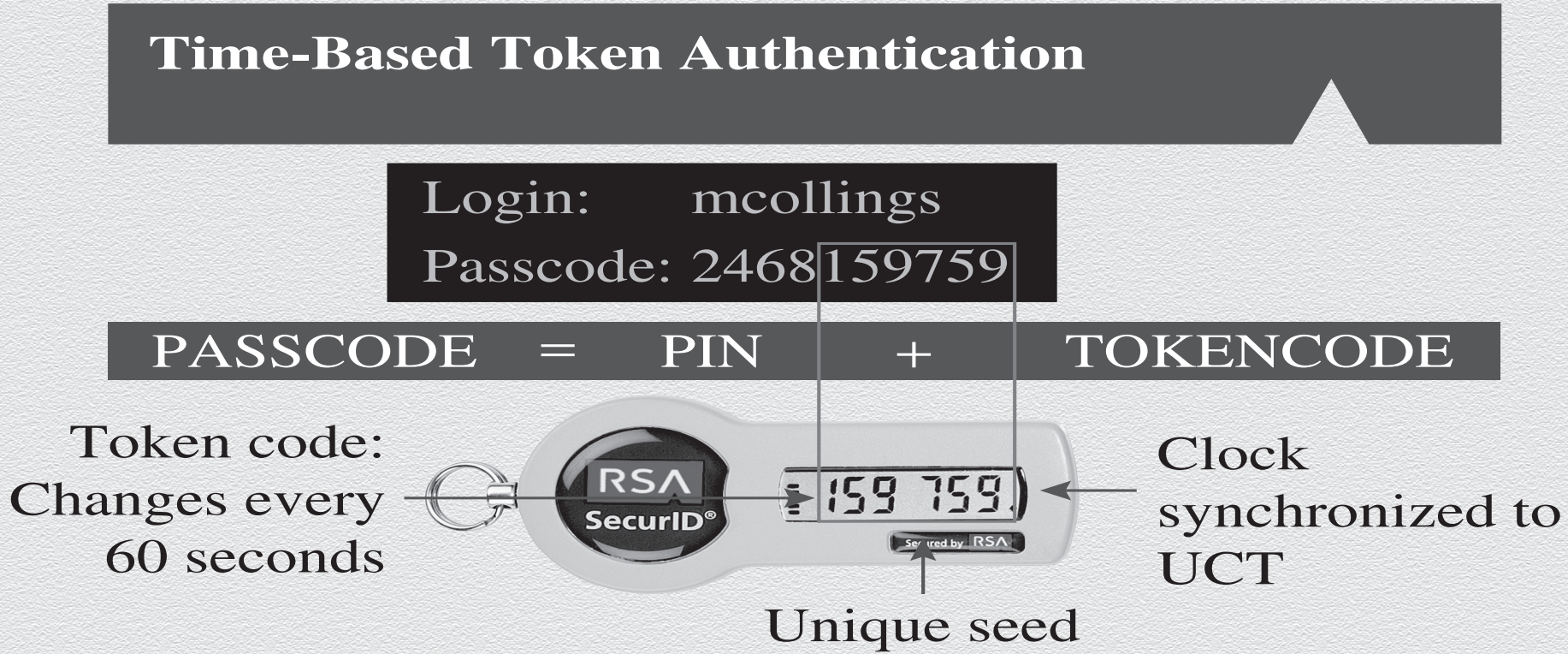
- you are leaving your fingerprints in many places
- rubber fingers have defeated commercial fingerprint-recognition
- minor issue if authentication takes place in the presence of security personnel
  - when authenticating remote users additional precautions have to be taken
- user acceptance: so far fingerprints have been used for tracing criminals

7.3 Something you have – authentication tokens

Something you have

- user presents a physical token to be authenticated
  - keys, cards or identity tags (access to buildings), smart cards
- limitations
  - physical tokens can be lost or stolen
  - anybody in possession of token has the same rights as legitimate owner
- physical tokens are often used in combination with something you know
  - e.g. bank cards come with a PIN or with a photo of the user
  - this is called: **2nd-factor authentication or multi-factor authentication**

Tokens: something you have

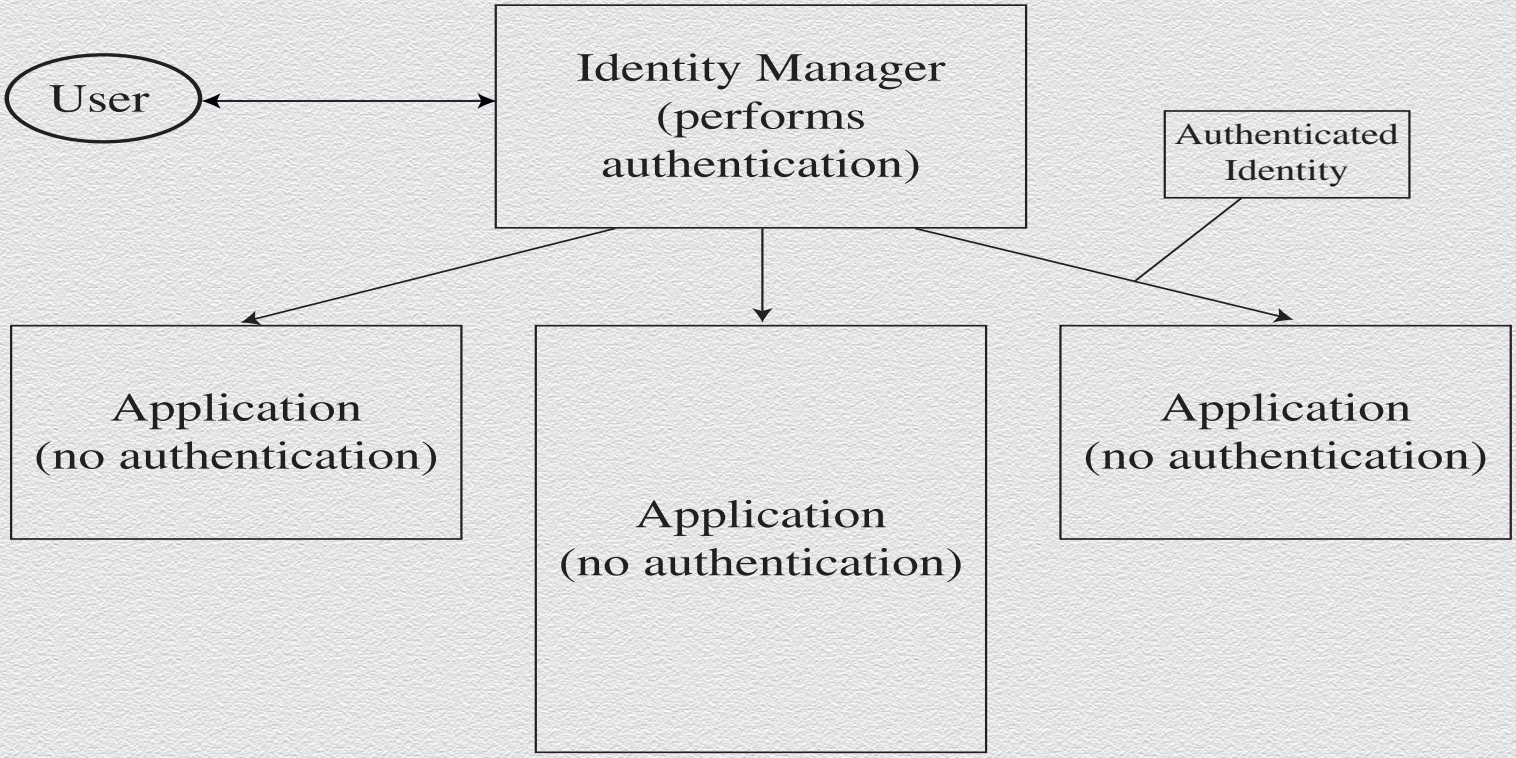


Problems with tokens

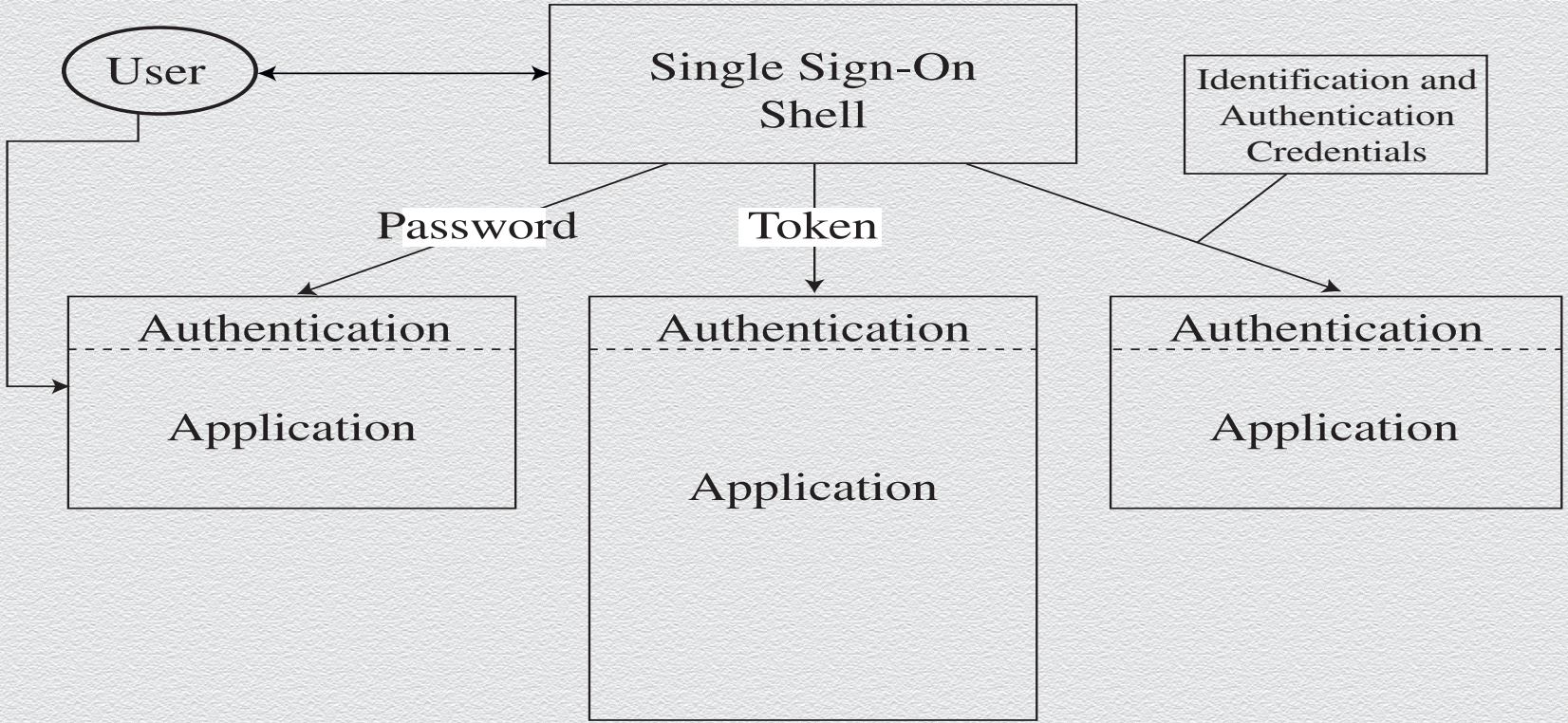
- Inconvenience
- Lost token
- Stolen token
- Cloned token
- Side-channel attacks (for key exfiltration)

7.4 Other user authentication methods

Federated identity management



SSO: Single Sign-On



More details on SSO

- Having to remember many passwords for different services is a nuisance
  - with a single sign-on service, you have to enter your password only once
  - (an alternative solution: password managers)
- A simplistic single-sign on service could store your password and do the job for you whenever you have to authenticate yourself
  - such a service adds to your convenience but it also raises new security concerns
- System designers have to balance convenience and security
  - ease-of-use is an important factor in making IT systems really useful
  - but many practices which are convenient also introduce new vulnerabilities

More on authentication

If dissatisfied with security level provided by passwords?

- you can be authenticated on the basis of
  - something you know
  - something you have
  - something you are
  - what you do – behavioural
  - where you are – location based

What you do

- people perform mechanical tasks in a way that is both repeatable and specific to the individual
- experts look at the dynamics of handwriting to detect forgeries
- users could sign on a special pad that measures attributes like writing speed and writing pressure
- on a keyboard, typing speed and key strokes intervals can be used to authenticate individual users
- more recently behaviours from one's mobile phone have been studied

Where you are

- some OSs grant access only if you log on from a certain terminal
  - a system administration may only log on from an operator console but not from an arbitrary user terminal
  - users may be only allowed to log on from a workstation in their office
- common method in mobile and distributed computing
- Global Positioning System (GPS) might be used to established the precise geographical location of a user during authentication