1495 UNIVERSITY OF ABERDEEN | CELEBRATING 525 YEARS 1495 – 2020

ABERDEEN 2040

Network Security Technology

Fundamentals of authentication

September 2025

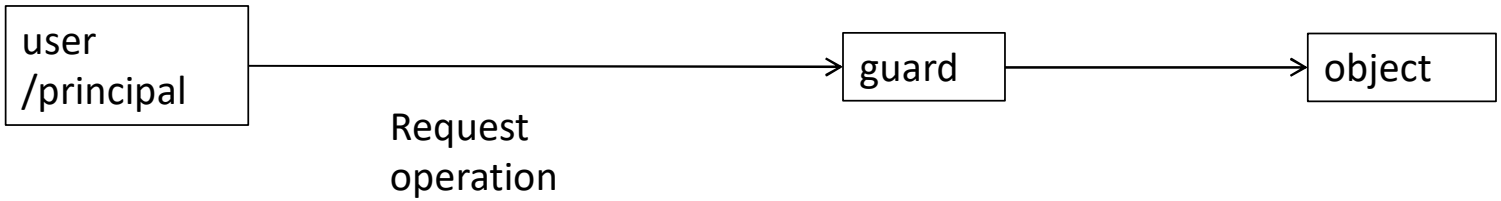
Outline of lecture

1. Motivation.
2. Access control and authentication processes.
3. Verification and proof of identity.
4. Factors and usability.
5. Trade-off: absolute security vs. usability.
6. Timing issues.
7. Soft defences, deterrence, user-enabled accountability.

Motivation

1. Motivation: control access to computational resources. Just the right entities should have access to just the right resources in just the right contexts/times.
2. When an entity makes a request, this requires verification that the three `right' conditions hold together. In particular, this requires verification that the entity is one of the right entities, as claimed. This claim and verification part is authentication.
3. Authentication must use some attributes of entities, e.g. identity.
4. We cover more detail on access control in another lecture. We focus on simple versions of authentication today (mostly for users). We discuss how the principal (user) proves their identity.

A user-oriented model of access control



1. A user makes a request for an access operation on a passive object (resource), possibly via some program. A guard (reference monitor) grants or denies access.
2. Two steps:
    1. **Authentication:**
        1. The process of verifying claimed attributes.
    2. **Authorization:**
        1. The process of establishing whether a request should be accepted or rejected, given successful authentication.

Complementary data

1. Guard checks **authentication data** supplied (e.g. password) against some **complementary data** about acceptable authentication data.
    1. The complementary data will often not be identical to the authentication data.
2. Authentication further divided into two steps:
    1. **Announcement:**
        1. Make claim of attributes.
    2. **Verification:**
        1. Also known as **authentication** (confusingly).

Claiming a user identity

1. **Entity authentication:** The process of verifying a claimed identity
    1. That is, the claimed attribute a user identity.
    2. In this case, announcement is called **identification,** and you announce your name.
        1. E.g. type in username.
    3. **Verification** is a proof that you are who you claim to be.
        1. E.g. supply your password.

Verification and proof of identity

1. How can you prove who you are?
2. It must be on the basis of some collection of **attributes** that you have, that the guard can understand and can use, and that you can (securely) communicate to them.
3. If you need to prove a unique identity, then the attributes must identify you uniquely.
4. Another way is for a trusted 3rd party to assure the guard that you are okay.
    1. This just delegates the authentication to the 3rd party.
    2. And can perhaps be passed further down a chain.

The 3 main authentication factors

A **factor** is a class of attributes for demonstration of authenticity. The main three are widely quoted:

1. Factor: **Something you know**
   - Attribute class: demonstration of knowledge
   - Proofs: e.g., a password, mother's maiden name.

2. Factor: **Something you have**
   - Attribute class : possession of an item (of some class)
   - Proofs: e.g. physical key, UoA ID card, smart card

3. Factor: **Something you are**
   - Attribute class: an intrinsic/inherent characteristic of you
   - Proofs: e.g., a fingerprint, iris scan or other biometric.

Further possible factors

1. People sometimes mention other possible factors. These often overlap with the 3 main factors to some degree.

For example, Gollmann mentions:

4. Factor: **What you do**
   - Attribute class : ability to produce an action
   - Proofs: e.g. generation of a (physical) signature.
5. Factor: **Where you are**
   - Attribute class: physical location
   - Proof: e.g. demonstration at a particular moment.

Stronger authentication

1. It may be that a single proof (as above) is highly usable, but too weak.
2. **Strong authentication**: guard asks for more than one proof.
3. **Two-factor authentication:** the guard insists that the user supplies two different factors correctly.
    1. Two proofs from the same factor do not count (e.g. knowledge of password and birthday).
    2. Common now with internet banking.
4. **Multi-factor authentication:** might insist on more than two factors.

Basic trade-off

1. It often turns out that a choice of authentication proof methods means taking into account a trade-off between the following two things.
2. **Absolute Security:**
    1. In the sense of difficulty of circumventing or beating the authentication system.
    2. Includes aspects of technical manageability.
        1. E.g. Levels of encryption and storage methods.
3. **Usability:**
    1. For individual users
    2. Poor usability can lead to users or management developing work-arounds that compromise absolute security.
    3. At a management level:
        1. Users have to be managed, enrolled, trained, correctly assigned to roles, etc.
        2. Credentials have to be issued, monitored, used, recovered (if lost), revoked.

Trade off

1. **Question for the designer**: What is a good system that allows for good overall security (right agents, right resources, right time)?
    1. How well does it prevent unauthorised access overall, and how easily does it allow authorised access?

Verification, repetition and timing

1. If user authentication fails, then you get bounced back to the login screen (usually).
    1. Sometimes there is a **backoff -** you have to wait longer each time you try and fail.
2. In some systems you only get a finite number of attempts before you are **locked out**.
    1. At this point you may have to go through a separate procedure, or wait a period of time before trying again.
3. **Repeat authentication:** You may have to re-authenticate at certain intervals during a session, not just at the beginning.
4. In computer security, you may be forced to re-authenticate if you do not interact with the machine for some time:
    1. e.g., the screen locks. This may also be done by user choice.

TOCTTOU

1. **TOCTTOU:** Time-of-check to time-of-use.
    1. The guard verifies the identity at one time, but often uses it for access decisions at some later time. There is a time interval between the two.
    2. This is true in many access decisions: by the OS on a single computer, by a server on a corporate network, or in the use of particular applications.
    3. There is a trade-off between vulnerabilities that can be exploited during this time interval and forcing users to re-authenticate repeatedly.

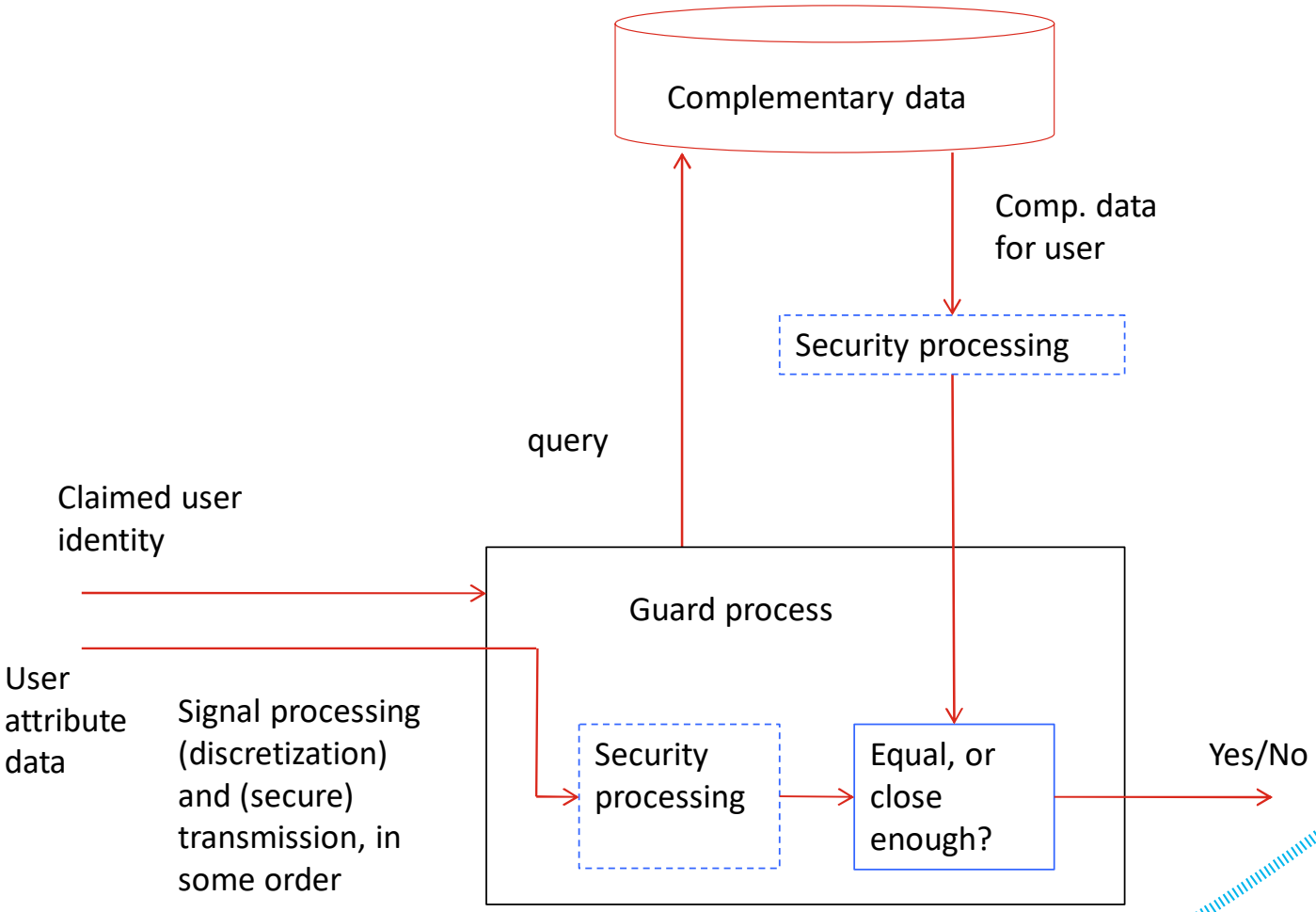
Verification with analogue, variable and imprecise attributes

1. Sometimes the attributes that are provided vary slightly over time. This is particularly the case with biometrics. It is not the case with passwords.
    1. E.g. If doing voice authentication, the signal from your voice will be different every single time.
2. You have to allow some threshold tolerances to deal with this fuzziness.
3. There can then be **false negatives**: the authentication system rejects genuine users.
4. There can be **false positives**: the authentication system verifies claims as correct where it should not.

Storing authentication data

1. The (authentication) guard is supplied with some attribute data by the user. It then checks this data against some stored data.
2. We call that stored data the **complementary data.**
3. The complementary data can be – and usually, when possible, should be – quite different to the authentication data supplied by the user.
4. This is true even taking into account any discretization and signal processing that may take place at either end of the transmission of the user supplied data.

Verifying authentication data with identities



Online vs. offline attacks

1. Is the attacker able to test the defences without actually launching a live attack?
2. **Online** (no):
    1. Example situation: password login, but complementary data not available or not usable.
    2. Method: guess directly. That is, repeatedly try password guesses until you find one (or some) that work.
    3. May result in lock-out, or discovery.
3. **Offline** (yes):
    1. Usually involves getting hold of the complementary data (e.g. password file).
    2. Example situation: Encrypted password file (complementary data) and obtained by attacker.
    3. Method: Test authentication data until find one input (password) that produces a match with an entry in the complementary data. Then use.
4. It is important to protect the complementary data.

A soft defence: deterrent

1. Immediately before and/or after authentication, give the user a warning that they shouldn't try to gain access if they are not authorised (in the non-technical sense, that is, that they are not intended to be able to gain access).
    1. Corporate windows systems often display some legal notice dialog box after login, that users then have to acknowledge to continue.
2. After successful login:
    1. Alert user about last time of login, and number of failed attempts since then.
    2. Give the user a way to report suspicious activity.
    3. **Users can be your friends**.

Further uses of the authentication point

1. Immediately after authentication, provide advice and warnings to users regarding security updates and patches that are required or recommended. Prompt users for action.
2. Provide users with information that is not security related.