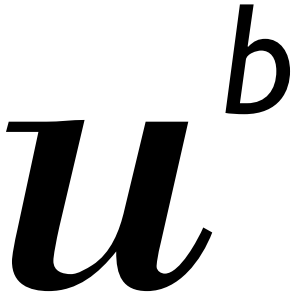


b

UNIVERSITÄT
BERN

UNIVERSITÄT
BERN

Network Security

VI. Authentication

Prof. Dr. Torsten Braun, Institut für Informatik

Bern, 28.03.2022 – 04.04.2022

UNIVERSITÄT BERN

Authentication

Table of Contents

1. Authentication Systems

- Authentication is the process of reliably verifying the identity of someone (or something).
  - Computers authenticate each other, e.g., printer and printer spooler.
  - A user must be authenticated when trying to use a computer system.
- Authentication types
  1. Password-based
  2. Address-based
  3. Cryptographic
- In many cases authentication leads to the establishment of a secret key between the communicating entities.

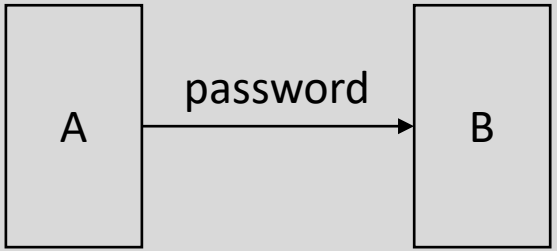
UNIVERSITÄT BERN

1. Authentication Systems

1. Password-based Authentication

- No cryptographic operation because:
  - Difficult to achieve by humans when connecting from dumb terminals
  - Crypto could be overly expensive in processing resources.
  - Export or legal issues
- Problems: Eavesdropping, cloning, etc.

  → Passwords should not be used in networked applications without additional protection.



UNIVERSITÄT BERN

1. Authentication Systems

1.1 Password-based Authentication: Password Guessing: On-line Attack

Operation

– try passwords until accepted

Protection

– limit number of trials and lock account: e.g., ATM machine; Denial-of-Service problem: lock all accounts

– increase minimum time between trials

– prevent automated trials: from a keyboard, Turing tests

– long passwords: pass phrases, initials of sentences, reject easy passwords

1. Authentication Systems

1.2 Password-based Authentication: Password Guessing: Offline Attack

Operation

- Attacker captures X = f(password)
- Dictionary attack: try to guess the password value offline
- Unix system uses a salt
  - Store “SS | hash(SS | password)”, where SS is a random value

Protection

- If offline attacks are possible, then the secret space should be large and should mitigate against GPU/ASICs parallel brute forcing.

1. Authentication Systems

1.3 Password-based Authentication: Storing Passwords

Alternatives

- Each user's secret information is stored in every server.
- Authentication Storage Node
  - stores user's secret information
  - Server has to retrieve user's secret information for authentication.
  - need to trust/authenticate/secure ASN session
- Authentication Facilitator Node
  - stores user's secret information
  - Server retrieves information from user and forwards it to AFN.
  - AFN does authentication and returns yes/no.
  - need to trust/authenticate/secure AFN session

Authentication Information Database

- Encryption
- Hashing, e.g., in UNIX, but allows offline attacks

UNIVERSITÄT BERN

1. Authentication Systems

2. Address-based Authentication

Access right is based on user@address.

Address-based authentication

- assumes that identity of the source can be inferred from network address of received packets.
- trusts network address information.
- is implemented in Unix, e.g., /etc/host.equiv, and .rhost files, NFS mounting
- is safe from eavesdropping.

Problems

- If an attacker gets access to a node, it also can get access to resources the node has access to.
- Impersonation of network addresses, e.g., IP or MAC addresses, is easy.
- Misuse of IP source routing

1. Authentication Systems

3.1 Cryptographic Authentication: Authentication Tokens

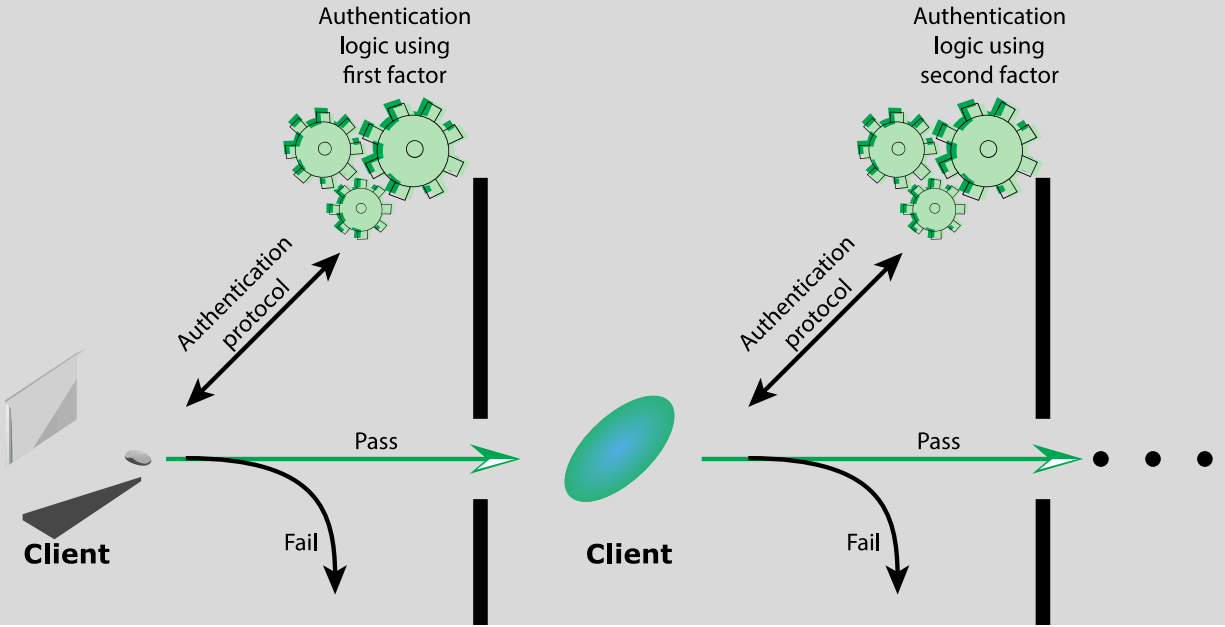
Authentication through what you have:

- Primitive forms: credit cards, physical key
- Smartcards-based: embedded CPU (tamper proof)
  - Personal Identification Number protected memory card: locks itself after a few wrong trials
  - Cryptographic challenge / response cards
    - Crypto key inside the card and not revealed even if given the PIN
    - Computer knowing the key in the card sends a challenge for encryption
    - If challenge is answered correctly, computer assumes availability of card and correct PIN
    - PIN authenticates the user (to the card), reader/server authenticates card
  - Cryptographic calculator: like previous card but has a display or speaker to output some result to be entered in a computer.
- Mobile devices as a second-factor

UNIVERSITÄT BERN

1. Authentication Systems

3.2 Cryptographic Authentication: Multifactor Authentication



1. Authentication Systems

3.3 Cryptographic Authentication: Passwords as Cryptographic Keys

- Cryptographic key: specially chosen large number, but difficult to remember
- Passwords can be remembered by humans used to support cryptographic authentication and other security mechanisms.
- How to derive keys from a password?
- Challenges
  - Users choose weak and short passwords.
  - If information on key becomes available to adversary
    → possible offline-attack
  - Key derivation from password should balance resources, e.g.,
    it is computationally expensive to convert passwords into RSA key.

UNIVERSITÄT BERN

1. Authentication Systems

3.4 Cryptographic Authentication: Passwords as Cryptographic Keys

- Symmetric encryption, e.g., AES: hashing of passwords to 128/256 bits
- RSA: private key must be designed more carefully and is based on selecting 2 prime numbers.
- Jeff Schiller: conversion of user password into public key pair
  - Such schemes often perform poorly.
  - Off-line password guessing attack possible

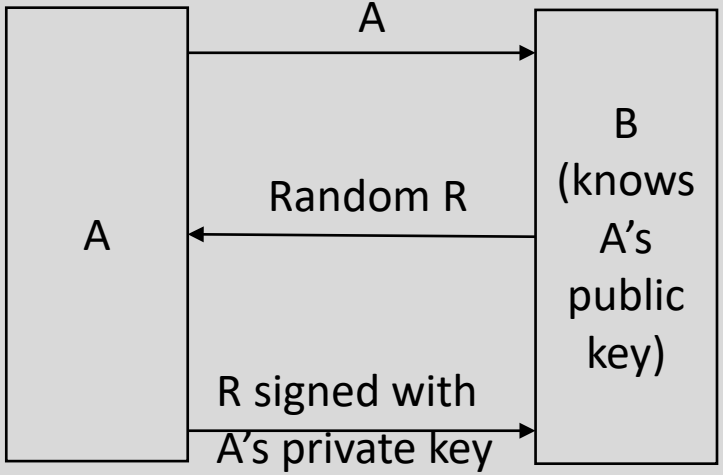
Method from Jeff Schiller

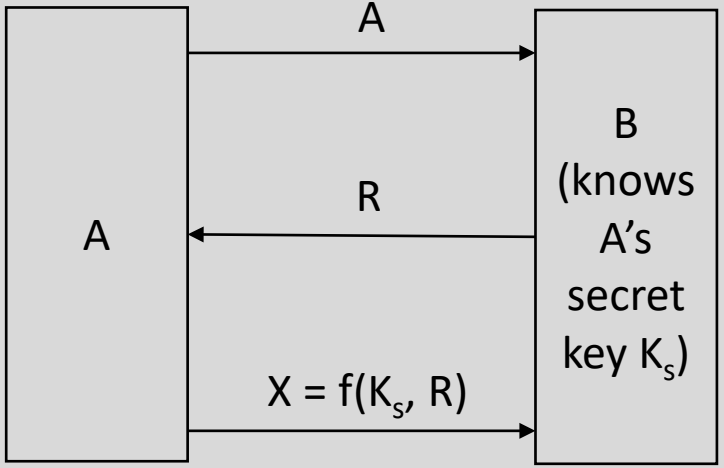
- Convert the password into a seed for a random number generator
- Random numbers must be tested whether being prime. (time consuming !)
- RSA key generator will find many non-primes before finding two primes.
- Example:
  - Key generator finds primes after 857 and 533 attempts using the user's seed.
  - Key generator can give <857,533> to user to remember and use for faster RSA key generation.

UNIVERSITÄT BERN

1. Authentication Systems

3.5 Authentication with Public and Secret Keys

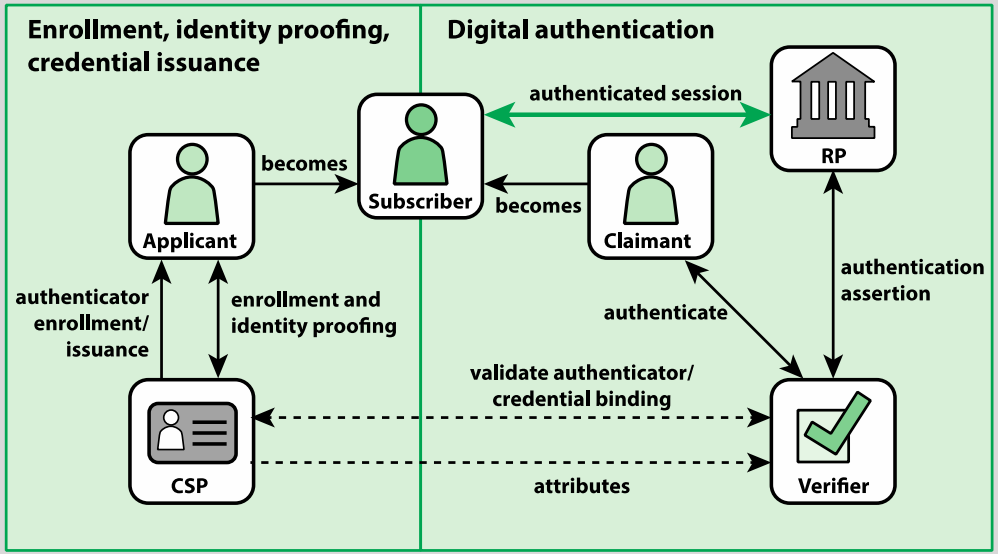




UNIVERSITÄT BERN

1. Authentication Systems

4.1 NIST Model for Electronic User Authentication



CSP = credential service provider
RP = relying party

UNIVERSITÄT BERN

1. Authentication Systems

4.2 NIST Model for Electronic User Authentication

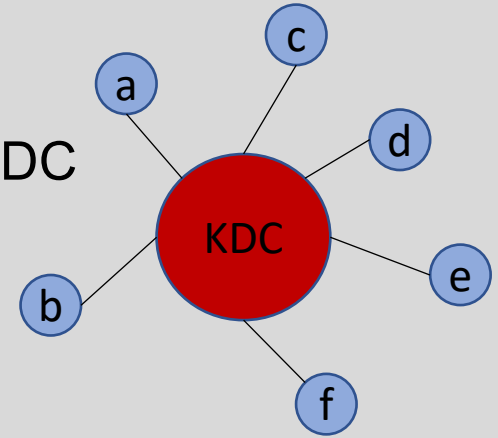
- **Credential Service Provider** is a trusted entity that issues or registers subscriber authenticators
- **Subscriber** is a party who has received a credential or authenticator from a CSP.
- **Applicant** is a subject undergoing the processes of enrollment and identity proofing.
- **Claimant** is a subject whose identity is to be verified using one or more authentication protocols.
- **Verifier** verifies the claimant's identity by verifying the claimant's possession and control of one or two authenticators using an authentication protocol.
- **Relying Party** relies upon the subscriber's authenticator(s) and credentials or a verifier's assertion of a claimant's identity, typically to process a transaction or grant access to information or a system.

UNIVERSITÄT BERN

1. Authentication Systems

5.1 Trusted Intermediaries: Key Distribution Center

- KDC knows keys for all nodes.
- Two nodes a and b aiming to talk to each other ask KDC for secret session key.
- KDC
  - authenticates a,
  - chooses random number R_{ab} as session key,
  - encrypts R_{ab} with both secret keys shared with a and b,
  - transmits encrypted R_{ab} to a and b, respectively.



UNIVERSITÄT BERN

1. Authentication Systems

5.2 Trusted Intermediaries: Certificate Authorities

- Public keys must be known, but it must be sure that they are correct and not overwritten by an attacker.
- CA generates certificates, which are signed messages of (name, public key).
- Certificates can be stored in a central repository or in a distributed way.

Advantages of CA

- CA does not need to be online.
- Failure of CA does not harm network operation but installing new devices / users.
- Certificates might be deleted, but alteration is difficult.
- Compromised CA can not decrypt messages, but compromised KDC can.

1. Authentication Systems

5.3 Trusted Intermediaries: Certificate Revocation

Problem

Certificates have a lifetime until when they are valid.
In case of earlier expiration, they have to be revoked.

Solution:
Certificate Revocation Lists

List of certificates (serial numbers) that should not be honored any more

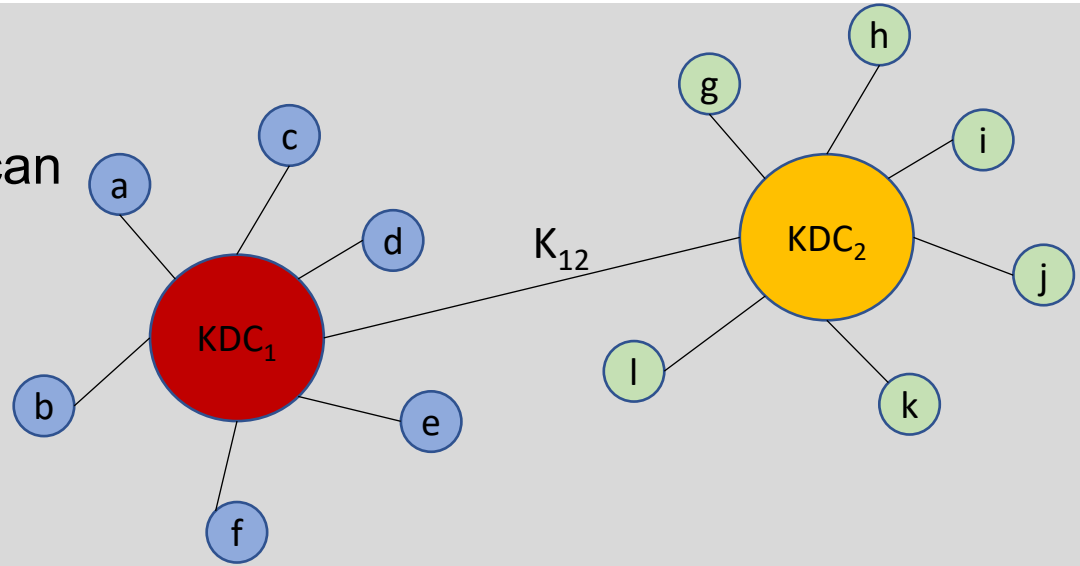
UNIVERSITÄT BERN

1. Authentication Systems

5.4 Multiple Trusted Intermediaries

Multiple KDCs

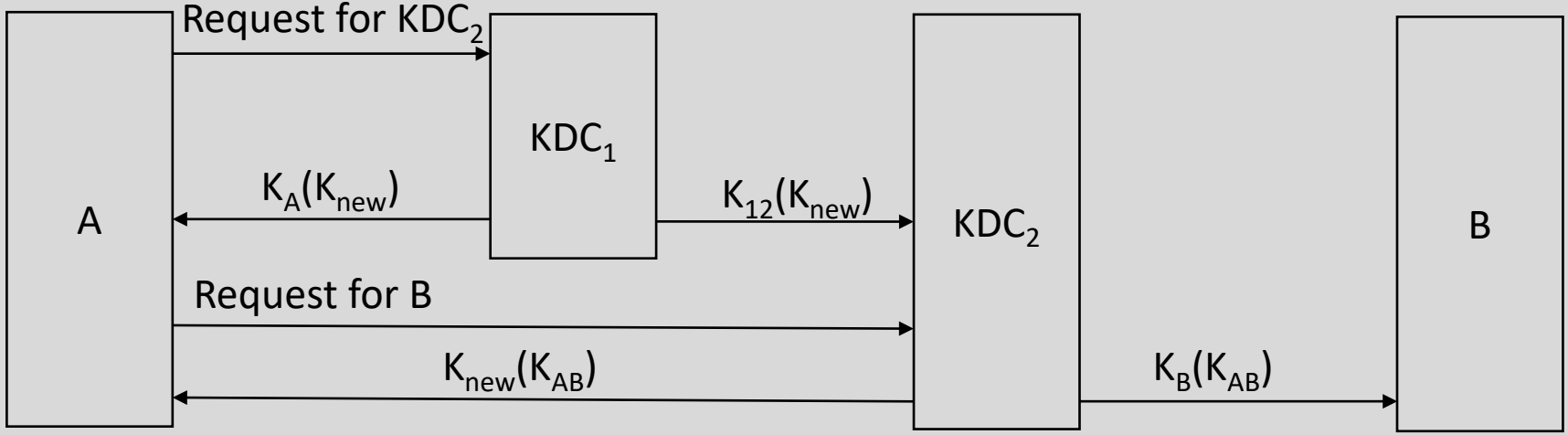
- to avoid that a single KDC can be compromised
- to improve scalability



UNIVERSITÄT BERN

1. Authentication Systems

5.5 Trusted Intermediaries: Multiple KDC Domains



2. Authentication of People

1. Overview

- Humans are not able to securely store high-quality cryptographic keys.

Factor	Examples	Properties
Knowledge	User ID Password PIN	Can be shared Many passwords easy to guess Can be forgotten
Possession	Smart Card Electronic Badge Electronic Key	Can be shared Can be duplicated (cloned) Can be lost or stolen
Inherence	Fingerprint Face Iris Voice print	Not possible to share False positives and false negatives possible Forging difficult

- User authentication: Computer verifies that a user is what she/he claims to be.
  - What you know (knowledge)
  - What you have (possession)
  - What you are (inherence)

UNIVERSITÄT BERN

2. Authentication of People

2. Initial Password Distribution

Physical contact:

- go to the system admin, show proof of identity, and set password
- Drawback: inconvenient, security threats when giving the user access to the system admin session to set the password

Choose a random strong initial password (pre-expired password) that can only be used for the first connection

2. Authentication of People

3. Biometrics

- Retina Scanner
- Fingerprint readers
- Face recognition
- Iris scanner
- Handprint readers
- Voiceprints
- Keystroke timing
- Signatures

UNIVERSITÄT BERN

3. Authentication Protocols

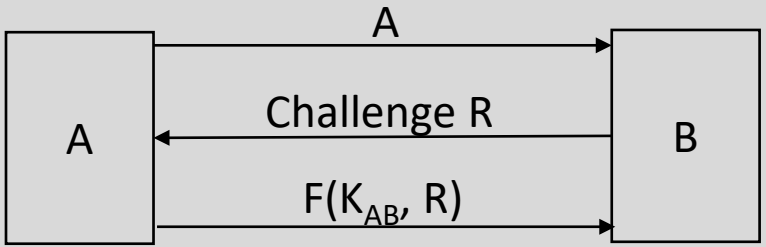
1.1 Password-based Protocols for Logins

1)

Simple password-based authentication:

- A sends password to B (clear ! or encrypted)
- B verifies password

2)

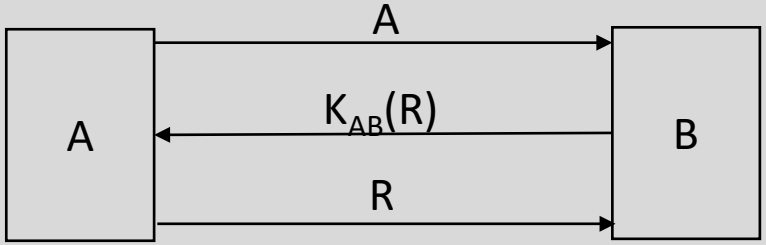


- More secure than simple algorithm
- No mutual authentication (A does not authenticate B), useful for logins
- Hijacking of conversation possible, i.e., generating packets with A's source address
- Off-line password attack possible

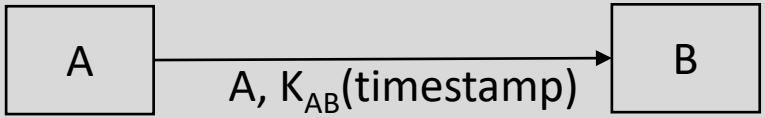
UNIVERSITÄT BERN

3. Authentication Protocols

1.2 Password-based Protocols for Logins



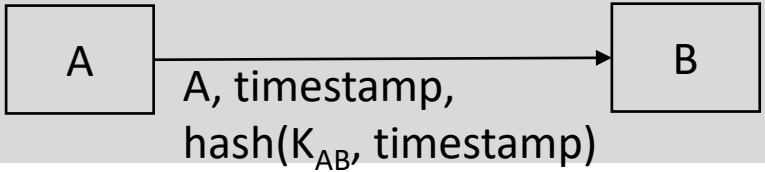
– Possible dictionary attack by sending requests or eavesdropping



– Authentication based on reasonably synchronized clocks

– Timestamps to be remembered to avoid impersonation

– B must verify for all possible timestamps

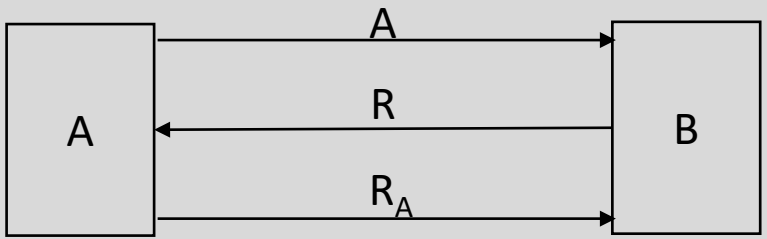


UNIVERSITÄT BERN

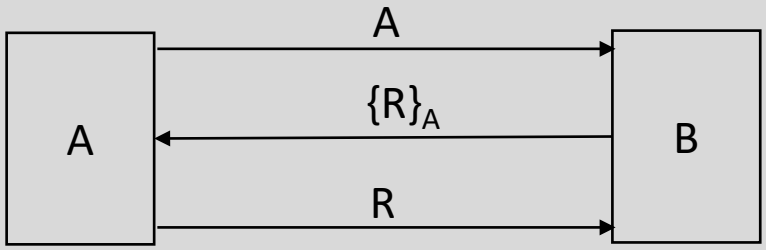
3. Authentication Protocols

2. One-Way Public Key

- Previous protocols allow impersonation, if user's database can be read.
- This can be avoided by public keys.
- Signature R_A: transform of R using A's private key



- Problem: one can trick someone to sign / decrypt something.
- Solution: never use R twice, e.g., by using a certain structure.
- $\{R\}_A$: R encrypted using A's public key

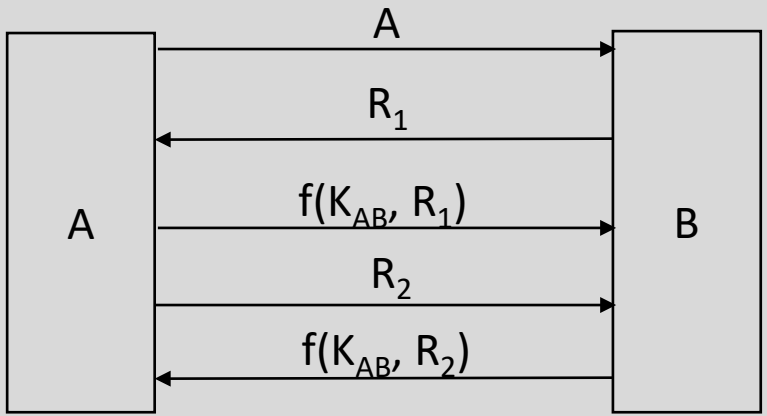


UNIVERSITÄT BERN

3. Authentication Protocols

3.1 Mutual Authentication

Mutual authentication based on shared key is rather inefficient.

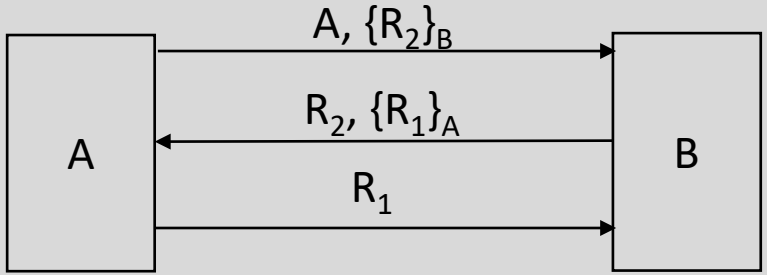


UNIVERSITÄT BERN

3. Authentication Protocols

3.2 Mutual Authentication

Mutual authentication based on public keys, but public keys must be known and verified, which could be a problem, if one party is a human.

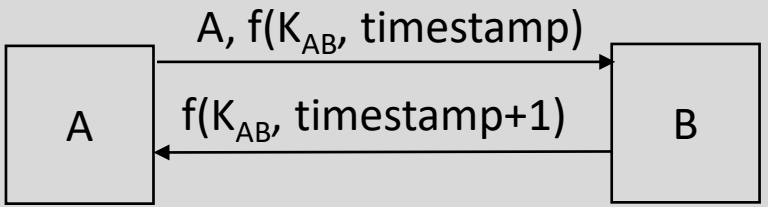


UNIVERSITÄT BERN

3. Authentication Protocols

3.3 Mutual Authentication

Mutual authentication with 2 messages
using timestamps and synchronized clocks

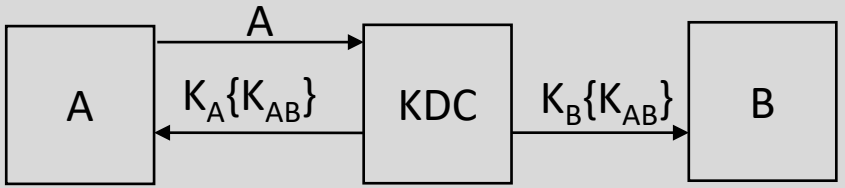


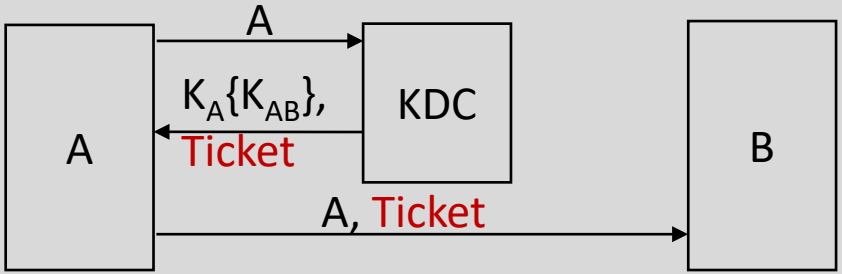
UNIVERSITÄT BERN

3. Authentication Protocols

4. Mediated Authentication

1. KDC operation in principle
   - A might receive message from KDC much earlier than B or vice versa.
2. KDC operation in practice
   - Ticket: $K_B\{A, K_{AB}\}$



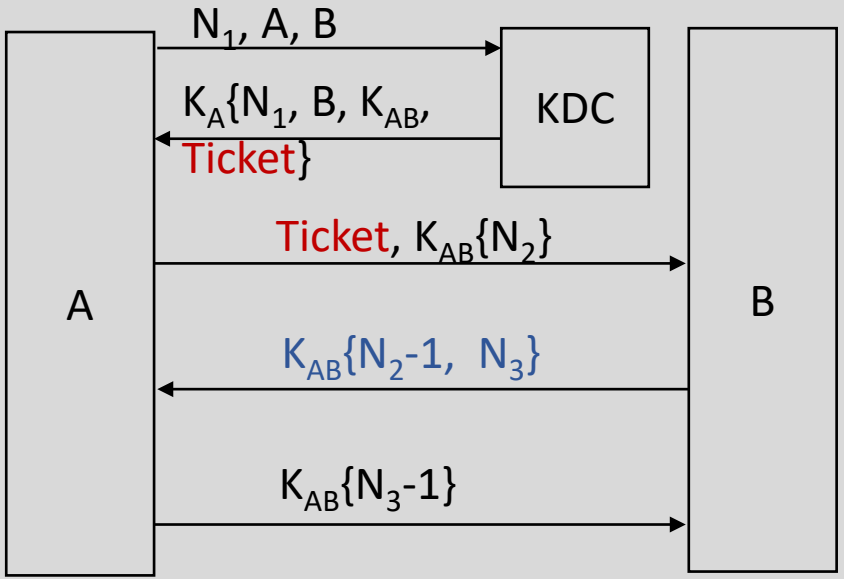


UNIVERSITÄT BERN

3. Authentication Protocols

5. Needham-Schroeder

- Nonces N_i
- Ticket: $K_B\{K_{AB}, A\}$
- Reflection attack possible in message 4, if symmetric encryption in ECB mode is used and nonces are separately encrypted.



3. Authentication Protocols

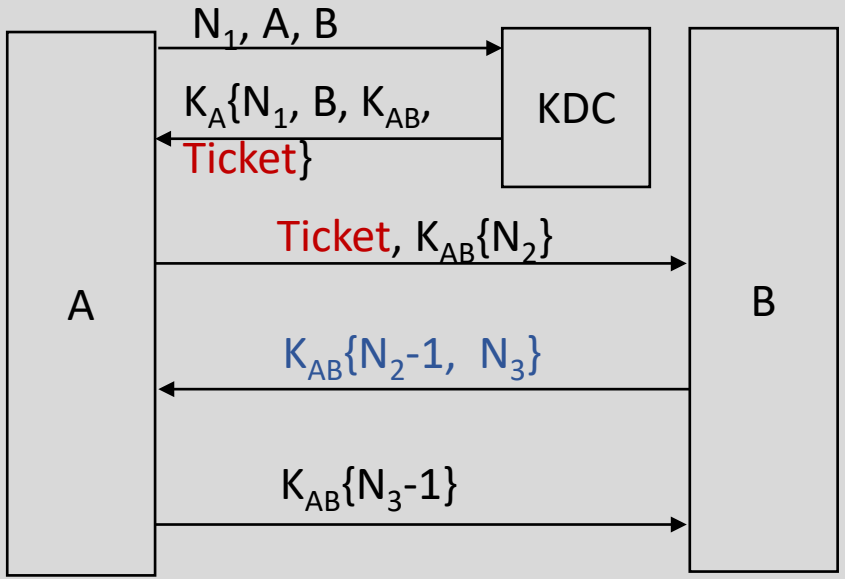
5.1 Reflection Attack

- Attacker initiates a connection to a target.
- Target attempts to authenticate the attacker by sending a challenge.
- Attacker opens another connection to the target and sends to the target this challenge as its own.
- Target responds to the challenge.
- Attacker sends that response back to target on original connection.

3. Authentication Protocols

5.2 Example: Reflection Attack

- N_2-1, N_3 are separately encrypted.
- C wants to impersonate A to B.
- C replays message 3 to B.
- B responds with $K_{AB}\{N_2\text{-}1, N_4\}$.
- C opens a new connection to B using $K_{AB}\{N_4\}$ instead of $K_{AB}\{N_2\}$, cf. message 3
- B returns $K_{AB}\{N_4\text{-}1, N_5\}$.
- C uses $K_{AB}\{N_4\text{-}1\}$ as message 5 of its connection.

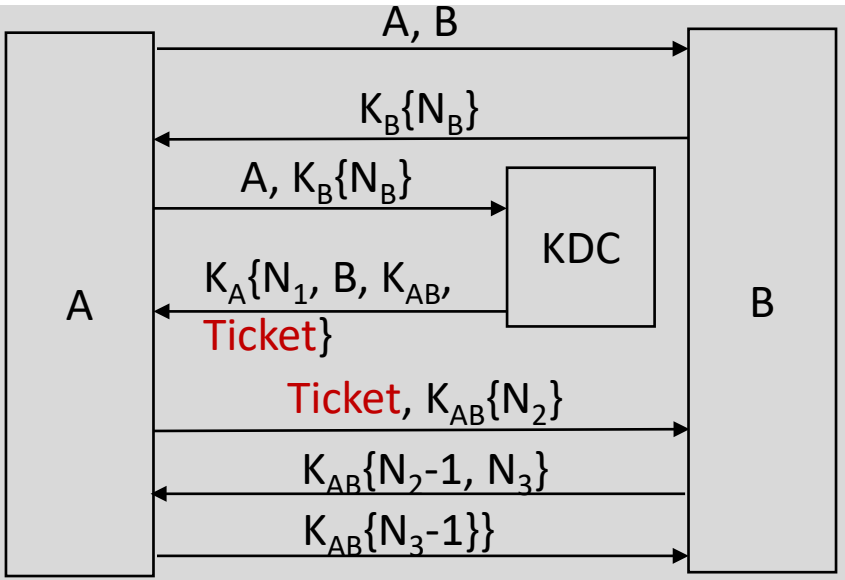


UNIVERSITÄT BERN

3. Authentication Protocols

5.3 Expanded Needham-Schroeder

- Problem of Needham-Schroeder:
  Attacker gets A's key:
  Ticket for B remains valid,
  even if A changes its key.
- Solution:
  Expanded Needham-Schroeder
  - Ticket: $K_B\{K_{AB}, A, N_B\}$
  - Ticket will reassure that an entity has contacted KDC before contacting B

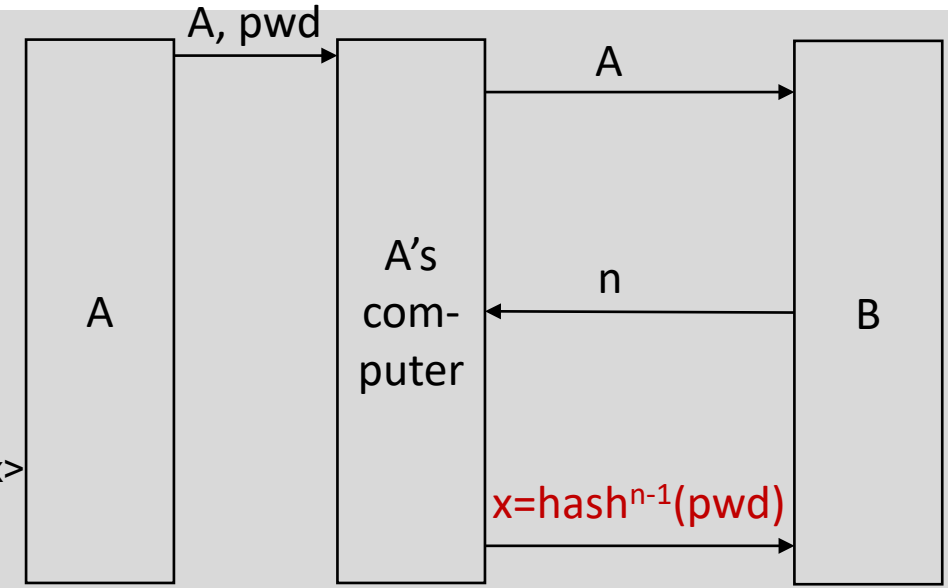


UNIVERSITÄT BERN

3. Authentication Protocols

6.1 Strong Password Protocols: Lamport's Hash

- B can authenticate A securely.
- Human A remembers a password.
- Server B has a database with user entries
  - user name
  - large n, e.g., 1000, which is decremented when a user authenticates
  - knows (hash^n(password))
- After receiving message: server
  - compares hash(x) with hash^n(password) and if equal, it replaces <n, hash^n(password)> by <n-1, x>
- Problems
  - Limited number of logins (n)
  - No mutual authentication

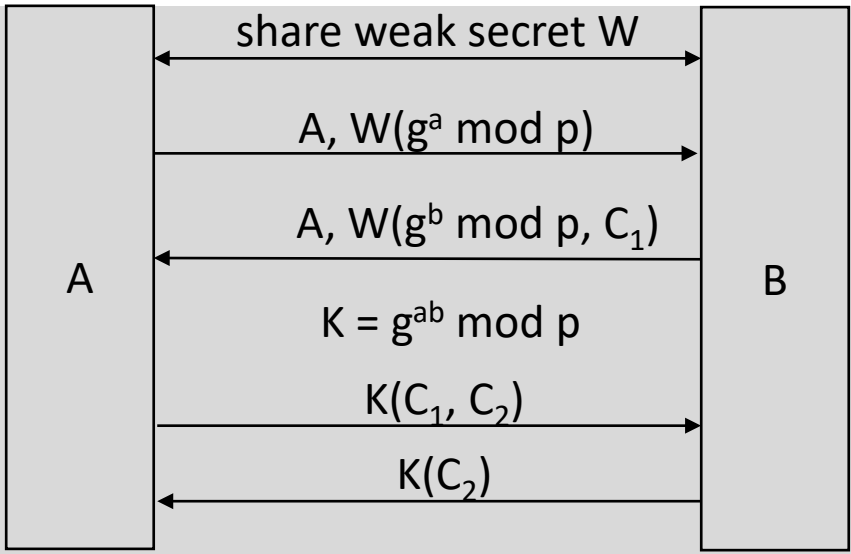


UNIVERSITÄT BERN

3. Authentication Protocols

6.2 Strong Password Protocols: Encrypted Key Exchange

- A and B share a weak password W, which is a hash of A's password; B stores it, A computes it.
- Diffie-Hellman exchange by encrypting DH numbers with W
- Attacker doing trial decryption fails, since decryption looks like a random number.
- Strong secret K, because attacker must guess password and break DH



4. Kerberos

1. Overview

- Authentication service developed as part of Project Athena at MIT
- A workstation cannot be trusted to identify its users correctly to network services, because
  - a user may gain access to a particular workstation and pretend to be another user operating from that workstation.
  - a user may alter the network address of a workstation so that the requests sent from the altered workstation appear to come from the impersonated workstation.
  - a user may eavesdrop exchanges and use a replay attack to gain entrance to a server or to disrupt operations.

Kerberos

- provides a centralized authentication server whose function is to authenticate users to servers mutually.
- relies exclusively on symmetric encryption, making no use of public-key encryption.

4. Kerberos

2.1 Replay Attacks

- An attacker simply copies a message and replays it later.
- An attacker can replay a timestamped message within the valid time window.
- An attacker can replay a timestamped message within the valid time window, but in addition, the attacker suppresses the original message.

4. Kerberos

2.2 Approaches against Replay Attacks

- Attach a sequence number to each message used in an authentication exchange
  - A new message is accepted only if its sequence number is in the proper order.
  - Difficulty with this approach is that it requires each party to keep track of the last sequence number for each claimant
  - Generally, not used for authentication and key exchange because of overhead
- Timestamps
  - require clock synchronization
  - Party A accepts a message as fresh only if the message contains a timestamp that, in A's judgment, is close enough to A's knowledge of current time.
- Challenge/Response
  - Party A, expecting a fresh message from B, first sends to B a nonce (challenge) and requires that subsequent message (response) from B contains correct nonce value.

4. Kerberos

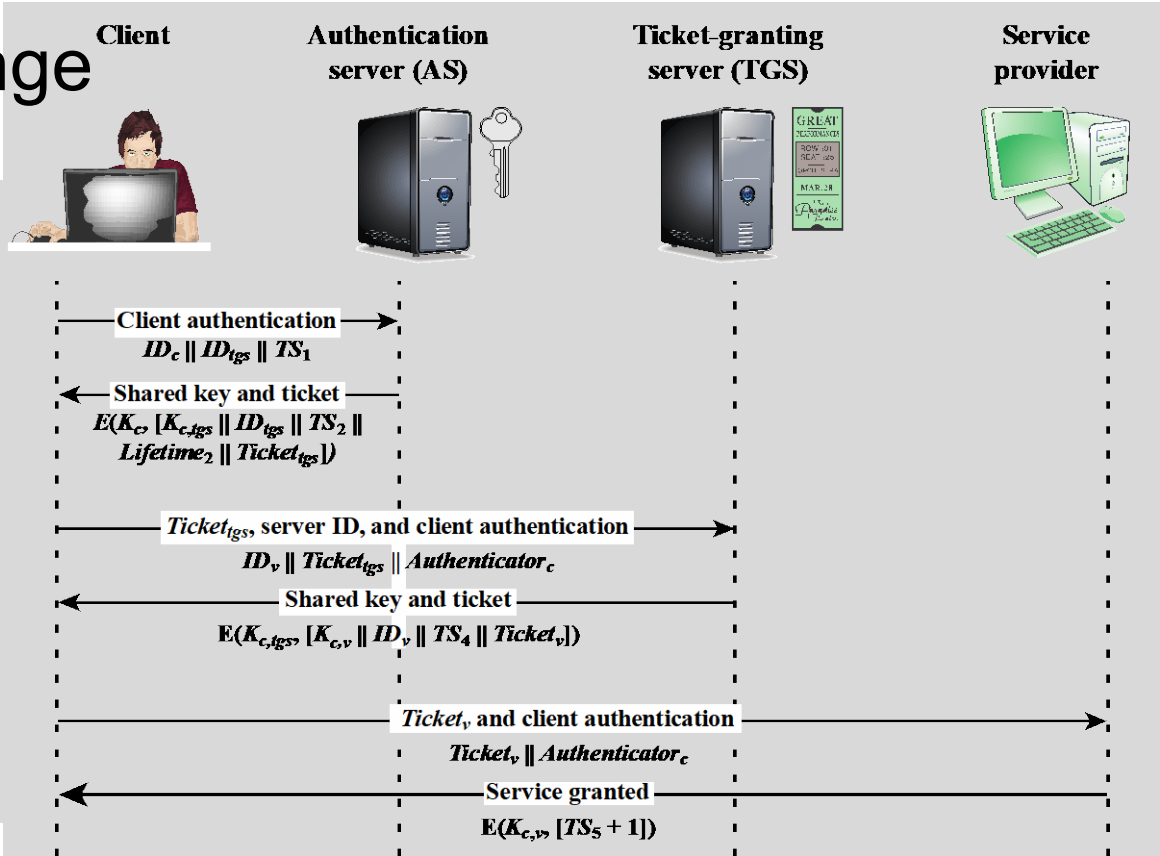
3. Kerberos V4

- uses DES for authentication service
- Authentication Server
  - knows passwords of all users and stores them in a centralized database
  - shares unique secret key with each server
- Ticket
  - is created once the AS accepts the user as authentic; contains user's ID, network address, server's ID
  - is encrypted using the secret key shared by AS and server
- Ticket-Granting Server
  - issues tickets to users who have been authenticated to AS
  - Each time the user requires access to a new service the client applies to TGS using the ticket to authenticate itself.
  - TGS then grants a ticket for service.
  - Client saves each service-granting ticket and uses it to authenticate its user to a server each time a particular service is requested

UNIVERSITÄT BERN

4. Kerberos

4. Message Exchange



4. Kerberos

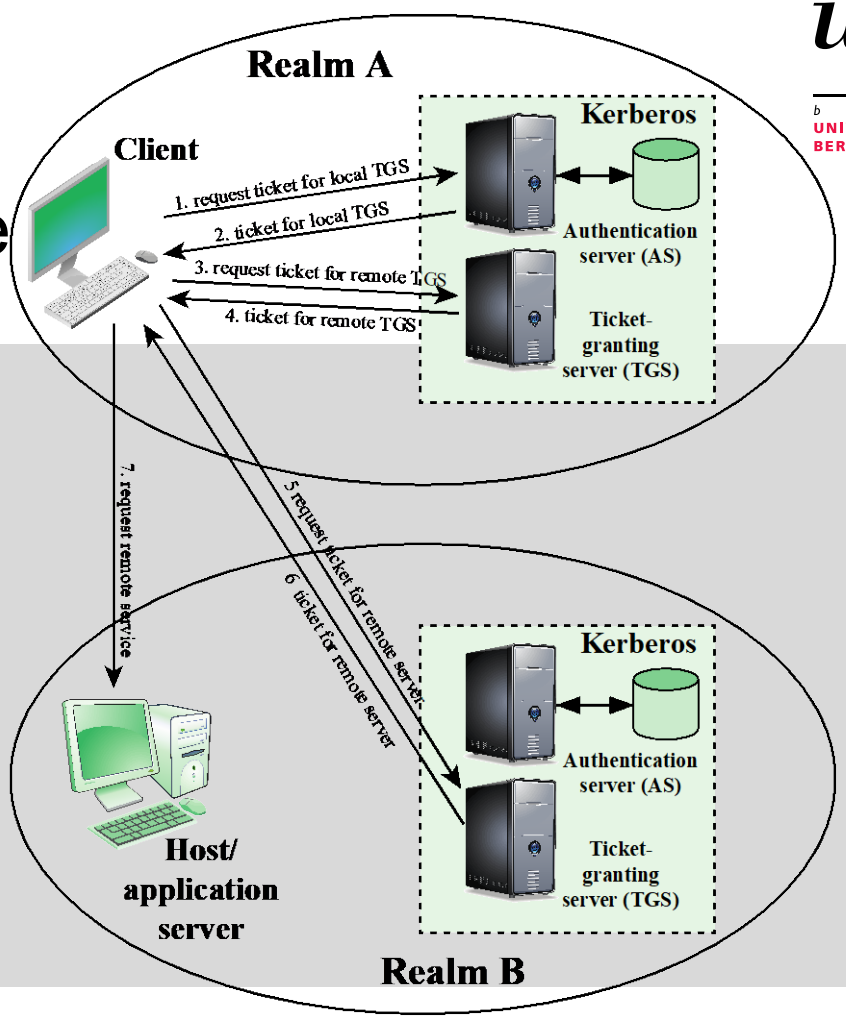
5. Realms and Multiple Kerberi

A full-service Kerberos environment, consisting of a Kerberos server, several clients, and several application servers requires that

- Kerberos server
  - must have user ID and hashed passwords of all participating users in its database.
  - must share a secret key with each other server.
  - in each interoperating realm shares secret keys with Kerberos servers in other realms.
- Realm = a set of managed nodes that share the same Kerberos database
- Mutual registrations

4. Kerberos

6. Request for Service in another Realm



4. Kerberos

7.1 Kerberos V4/5: Environmental Shortcomings

1. **Encryption system dependence**: V4 requires DES (export restrictions). V5 makes use of AES.
2. **IP dependence**: V4 requires IP addresses. V5 network addresses are tagged with type and length, allowing any network address types
3. **Message byte ordering**: In V4, the sender of a message employs a byte ordering of its own choosing. In V5, message structures are defined using Abstract Syntax Notation One (ASN.1) and Basic Encoding Rules.
4. **Ticket lifetime**: Lifetime values in V4 are encoded in 8-bits with units of 5 minutes, i.e., a maximum lifetime of $2^8 * 5 = 1280$ minutes. In V5, tickets include explicit start time and end time, allowing arbitrary lifetimes.
5. **Authentication forwarding**: V4 does not allow credentials issued to one client to be forwarded to some other host and used by some other client. For example, a client issues a request to a print server, which then accesses the client's file from a file server, using the client's credentials for access. V5 provides this capability.
6. **Inter-realm authentication**:
   In V4, interoperability among N realms requires on the order of N^2 Kerberos-to-Kerberos relationships.
   V5 supports a method that requires fewer relationships.

4. Kerberos

7.2 Kerberos V4/5: Technical Deficiencies

1. **Double encryption**: Tickets provided to clients are encrypted twice, which is not necessary.
2. Encryption in V4 makes use of Propagating Cipher Block Chaining, which is vulnerable to an attack involving the interchange of ciphertext blocks.
   V5 provides explicit integrity mechanisms, allowing standard **CBC mode** for encryption
3. **Session keys**: Each ticket includes a session key. Because the same ticket may be used repeatedly to gain service from a particular server, there is the risk that an opponent will replay messages from an old session to the client or the server. In V5, it is possible for a client and server to negotiate a subsession key, which is to be used only for that one connection.
4. **Password attacks**: Both versions are vulnerable to password attacks.
   V5 does provide a mechanism known as pre-authentication, which should make password attacks more difficult.

UNIVERSITÄT BERN

5. Federated Identity Management

1. Overview

dealing with the use of a common identity management scheme across multiple enterprises and numerous applications and supporting many users

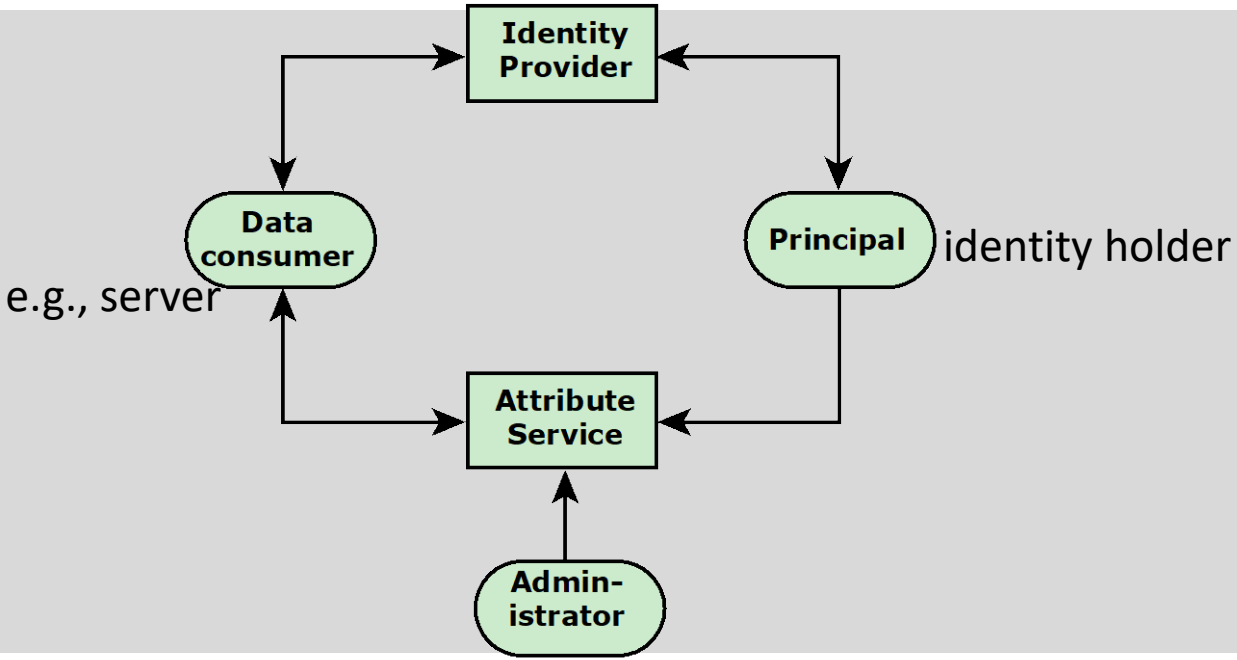
Services provided include:

- Point of contact
- Single-Sign-On protocol services
- Trust services
- Key services
- Identity services
- Authorization
- Provisioning
- Management

UNIVERSITÄT BERN

5. Federated Identity Management

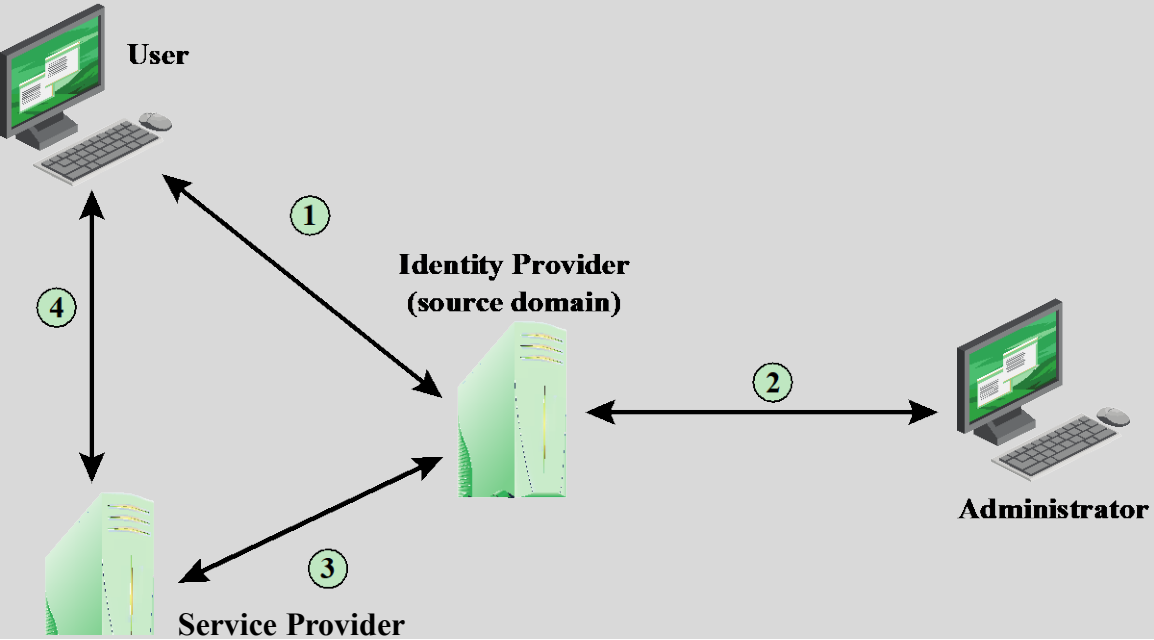
2. Generic Identity Management System



UNIVERSITÄT BERN

5. Federated Identity Management

3. Operation



1. User's browser or application contacts Identity Provider. End user provides attributes.
2. Administrator may also provide attributes.
3. Service Provider obtains identity information, authentication information, attributes from IdP.
4. SP opens session to user and enforces access control based on user's identity and attributes.

Thanks

for your Attention

Prof. Dr. Torsten Braun, Institut für Informatik

Bern, 04.04.2022 – 11.04.2022