DI-FCT-UNL
Segurança de Redes e Sistemas de Computadores
Network and Computer Systems Security

Mestrado Integrado em Engenharia Informática
MSc Course: Informatics Engineering
2º Semestre, 2018/2019

Authentication

Outline

- Authentication
- Common Authentication Protocols and Services
- Authentication approach levels: Multilayer Authentication
- Authentication Systems and Frameworks
- Federated Identity Management
- SSO with OpenID
- SSO with SAML

Outline

- Authentication
- Common Authentication Protocols and Services
- Authentication approach levels: Multilayer Authentication
- Authentication Systems and Frameworks
- Federated Identity Management
- SSO with OpenID
- SSO with SAML

Authentication

Authentication (in general):

Authentication is a process in which a **Proof of Authenticity** of a **Claimed Identity** is Presented (by a Claimant Principal) and Verified (by the Verifier), involving two steps:

- Identification (named principals or entities)
- Authentication (verification proof of claimed identity)

Two Steps (phases) involved:

- **Identification step**
  - **Presentation of valid (unique) identifier as a claimed identity**
- **Verification step**
  - Presenting or generating authentication information to corroborate the binding between the entity and the identifier.

Authentication as a Key Concern

Authentication is a key-concern for different requirements:

- Allowing for the correct enforcement of policies for permissions, access or authorization control, in verifying the proper permissions to use resources and to access data/information
  - Access Control Models over Authenticated Principals or Authenticated Principals in Roles
  - Keys distribution and establishment of security association parameters for secure communication channels

Authentication: Key concern for Access Control BUT different than Access Control (Authorization) !!! Two Separated Concerns !!!

Depending on the authentication challenge for identity claiming and purpose, valid credentials (with one or more specific factors as proofs) need to be exhibited

Authentication vs. Access Control

Identification and Authentication

1.	Alice to Bob:	Hi Bob, I am Alice	// Alice Claim		Identif. Step
2.	Bob to Alice:	Prove It	// Proof Challenge for Claim)		Auth. Step
3.	Alice to Bob:	Credentials of Alice	// Claim Proof		
4.	Bob to Alice:	Validated/Not Validated	// Claim Validation		
5.	Alice to Bob:	I want this …	// Autorization request	✗	Auth. (Acess Control)
6.	Bob to Alice:	Go Ahead Alice	// Authorization	✗	

Authentication Phases in Protocols and Directionality

Authentication Directionality

Unidirectional vs. Bidirectional (or mutual)

Unidirectional (or Unilateral)

1. Alice to Bob: Hi Bob, I am Alice // Alice Claim
2. Bob to Alice: Prove It // Proof Challenge for Claim)
3. Alice to Bob: Credentials of Alice // Claim Proof
4. Bob to Alice: Validated/Not Validated // Claim Validation

Bidirectional (or Mutual)

1. Alice to Bob: Hi Bob, Prove your are B // Bob Challenge Proof
2. Bob to Alice: Bob Credentials // Proof of Bob Claim)
3. Alice to Bob: Validated/Not Validated // Bob Claim Proof Validation
4. Bob to Alice: Prove you are Alice // Alice Challenge Proof
5. Alice to Bob: Alice Credentials // Alice Claim Proof
6. Bob to Alice: Validated/Not Validated // Alice Claim Proof Validation

Authentication Phases in Protocols and Directionality

Authentication Directionality

Unidirectional vs. Bidirectional (or mutual)

Unidirectional (or Unilateral) challenge/response Authentication

1.	Alice to Bob:	A	// Alice Claim
2.	Bob to Alice:	A, F(), Cb	// Proof Challenge for Claim)
3.	Alice to Bob:	F_{KA}(A, Cb+1)	// Claim Proof
4.	Bob to Alice:	F_{KA}^{-1} (A, Cb+1)	// Claim Validation

Bidirectional (or Mutual) Authentication

1.	Alice to Bob:	A, B, Ca, F1()	// Bob Challenge Proof
2.	Bob to Alice:	$F1_{KB}$ (B, Ca+1	// Proof of Bob Claim)
3.	Alice to Bob:	OK Bob	// Validated/Not Validated
4.	Bob to Alice:	B, A, Cb, F2()	// Alice Challenge Proof
5.	Alice to Bob:	$F2_{KA}$ (A, Cb+1)	// Alice Claim Proof
6.	Bob to Alice:	OK Alice	// Alice Claim Proof Validation

Outline

- Authentication
- Common Authentication Protocols and Services
- Authentication approach levels: Multilayer Authentication
- Authentication Systems and Frameworks
- Federated Identity Management
- SSO with OpenID
- SSO with SAML

Simple and Common Authentication Protocols

Simple, Common and Weak Approaches

- HTTP Basic Authentication. Why is it weak ? Discussion

PPP Based Authentication means Point-to-Point

- PAP, CHAP (RFC 1334 /1992, RFC 1994/1996); Unilateral Authentication Protocols (Authenticator not validated)
  - PAP : PPP Authentication Protocol
    - Send a pair <user, password> // in plaintext
  - CHAP : CHallenge-Response Authentication Protocol

    Aut -> U : authID, challenge

    U -> Aut : authID, MD5 (authID, pwd, response), Identity

    Aut -> U: authID OK / Not OK

Require Secure Channel

MS-CHAP (RFC 2433/1998, RFC 2759/2000)

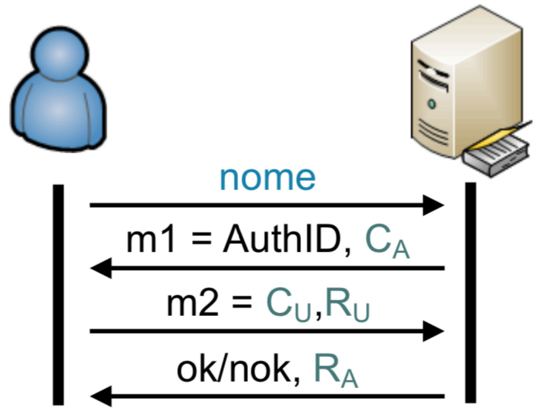
Version 1

```
A → U: authID, C
U → A: R
A → U: OK/not OK

R = DESPH (C)
PH = LMPH ou NTPH
      LMPH = DEShash(senha)
      NTPH = senha
```

Version 2

```
A → U: authID, CA            ← m1
U → A: CU , Ru               ← m2
A → U: OK/not OK, RA

RU = DESPH (C)
C = SHA(CU, CA, username)
PH = MD4(senha)
RA = SHA(SHA(MD4(PH), Ru, m1), C, m2)
```

MS CHAP v2:
Mutual Authentication: A, U
Possible Alteration of PWDs (*Senhas*)



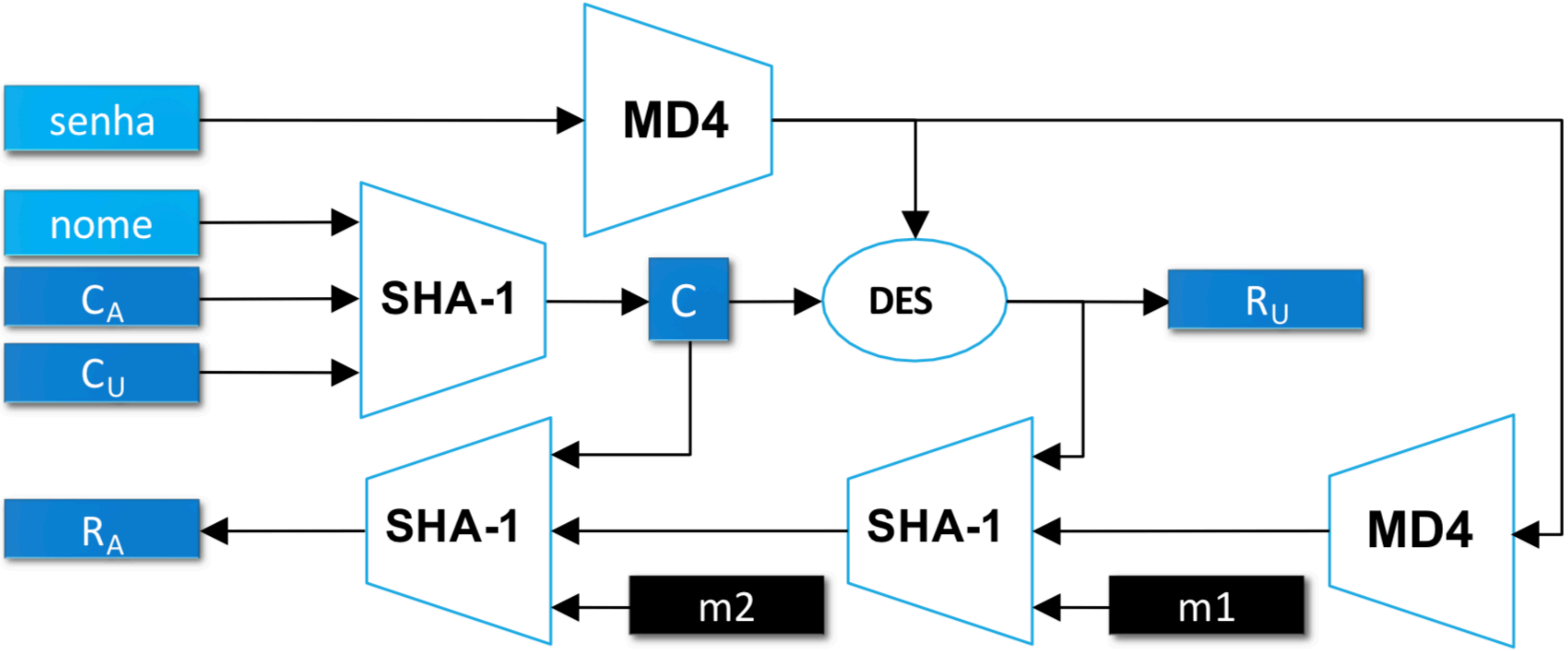
nome

m1 = AuthID, C_A

m2 = C_U,R_U

ok/nok, R_A

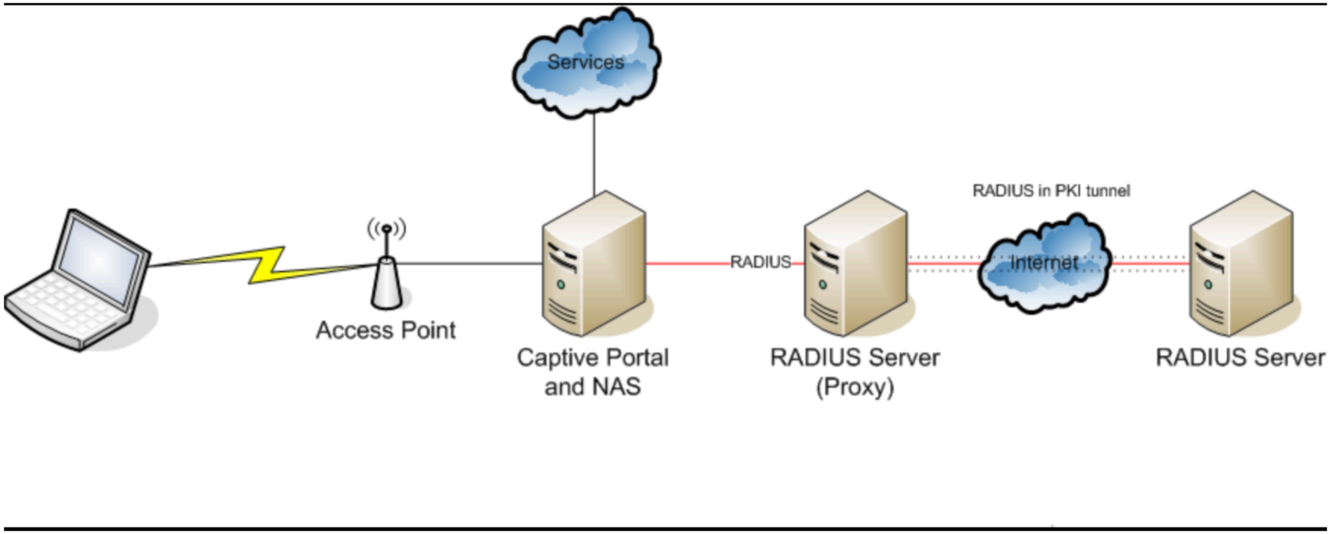
MS-CHAP v2



Could be enforced w/ Secure Channel ? Discussion !

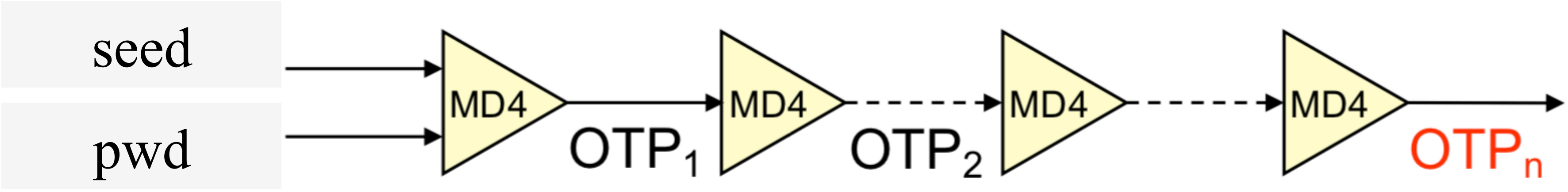
RADIUS (RFCs 2865 and 2866)

- Authentication, Authorization and Accounting (AAA) SSO service and protocol (using Link-layer PPP, for NAS-Devices <-> NAS (Network Access Server)
- Can use different Authentication Protocols: PAP, CHAP or EAP
- RADIUS Servers can also reuse external Authentication Services: Kerberos, SQL-DBs, LDAP and AD
- PWDs, sent obfuscated by MD5, but can also leverage from secure channels: ex., IPSec Tunnels or TLS channels)



See more: https://en.wikipedia.org/wiki/RADIUS

S-KEY (RFC 2289/1998)



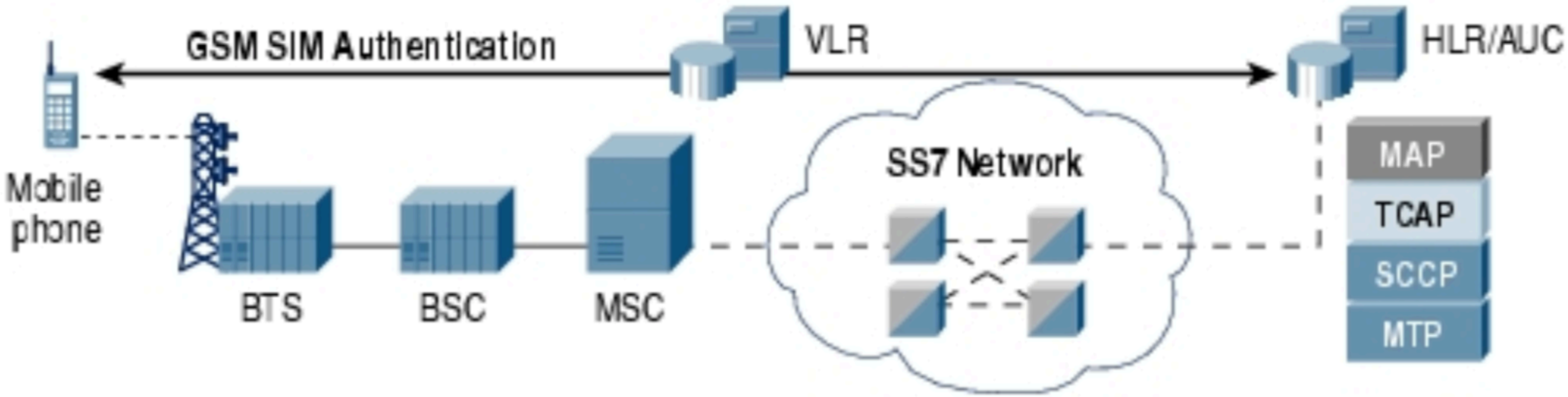
A -> U: seed + index i // as challenge

U generates i-1 consecutive OTPs
U selects the last OTP_{i-1} as the response for the challenge

U -> A: R = H (OTP_{i-1})

A: Computes H (R) and compares with OTP_i
If is the same, SUCCESS
if SUCCESS memorizes i-1 and OTP_{i-1}

GSM Authentication



Shared Secret Key: HLR and Mobile Phone

- 128bits Ki, stored in the SIM Card
- Only usable after the local PIN Authentication (to unblock the phone)

Algorithms (initially not public, GSM Consortium):

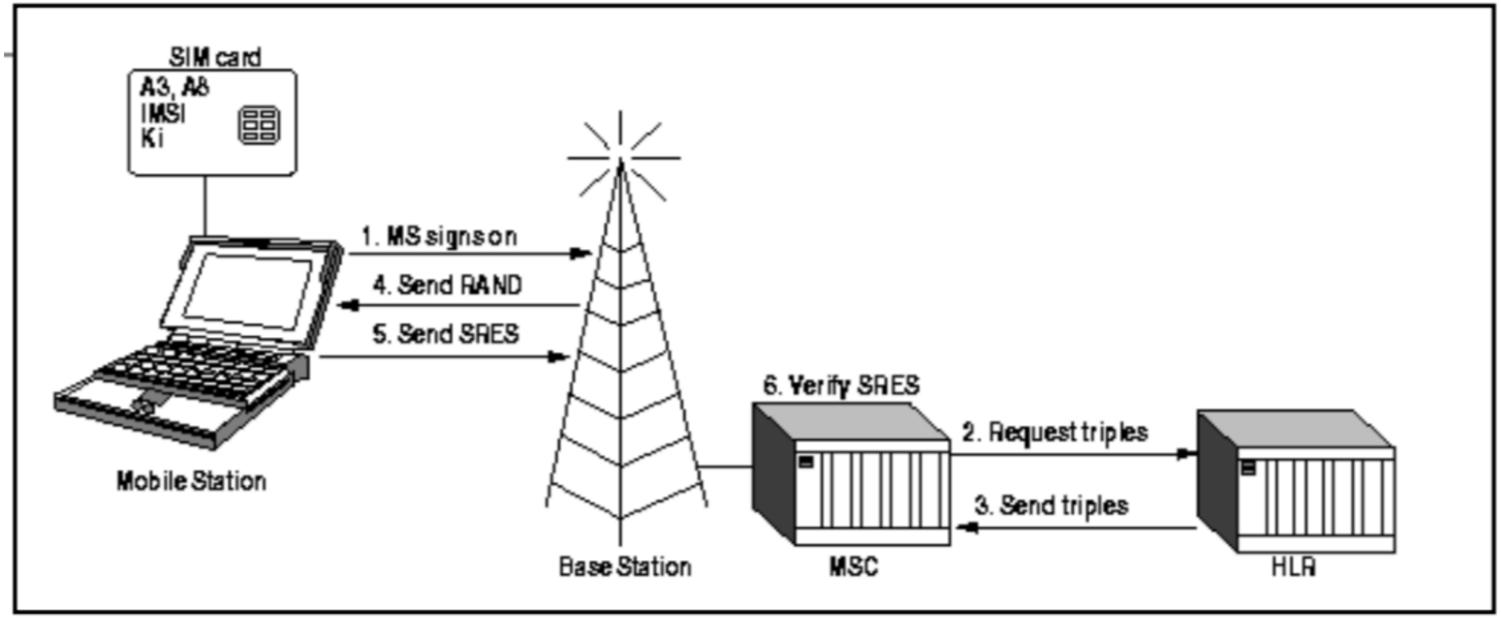
A3 for Authentciation

A8 for Key-Genaration

A5 for encryption (communication)

A3 and A8 (implemented inside the SIM card)

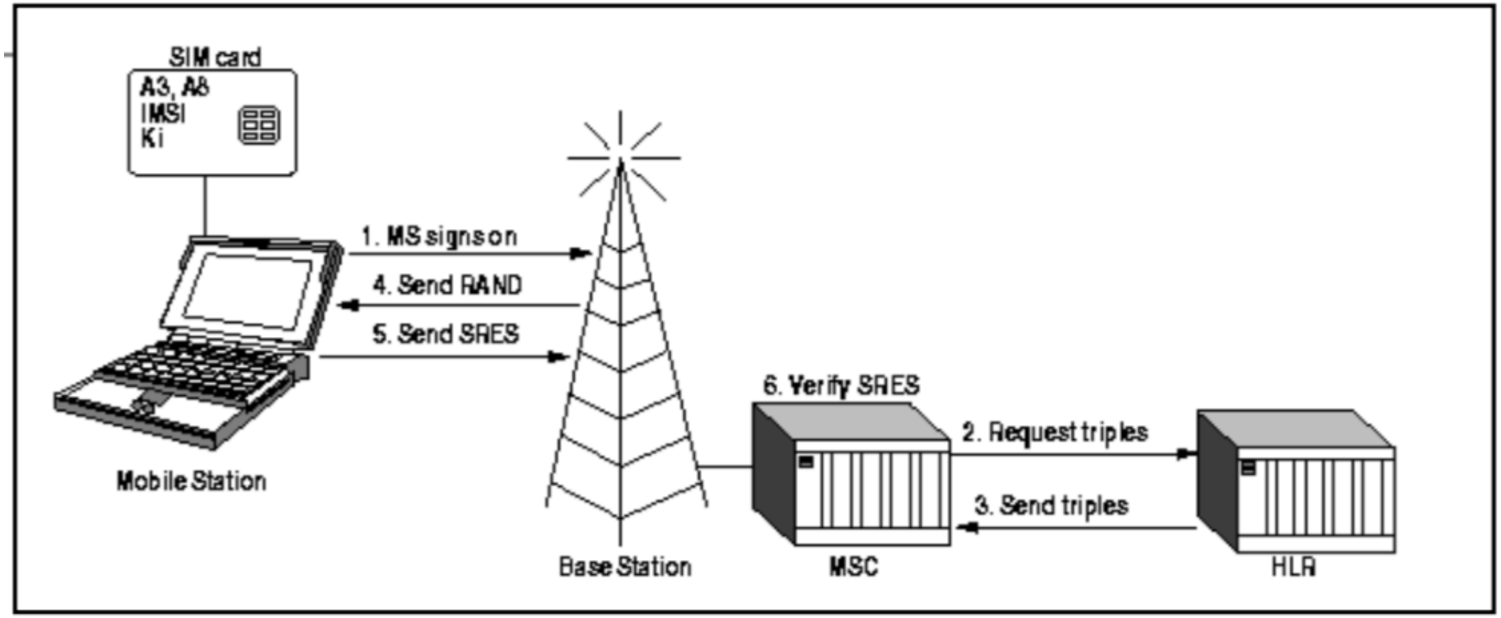
GSM Authentication Protocol



MSC asks **HLR** for triples
RAND, SRES, Kc

HLR generates RAND and a corresponding triple, using Ki form the subscriber

GSM Authentication Protocol



RAND (random number) : 128 bits)
SRES = A3 (Ki, RAND) : 32 bits
Kc = A8 (Ki, RAND) : 64 bits

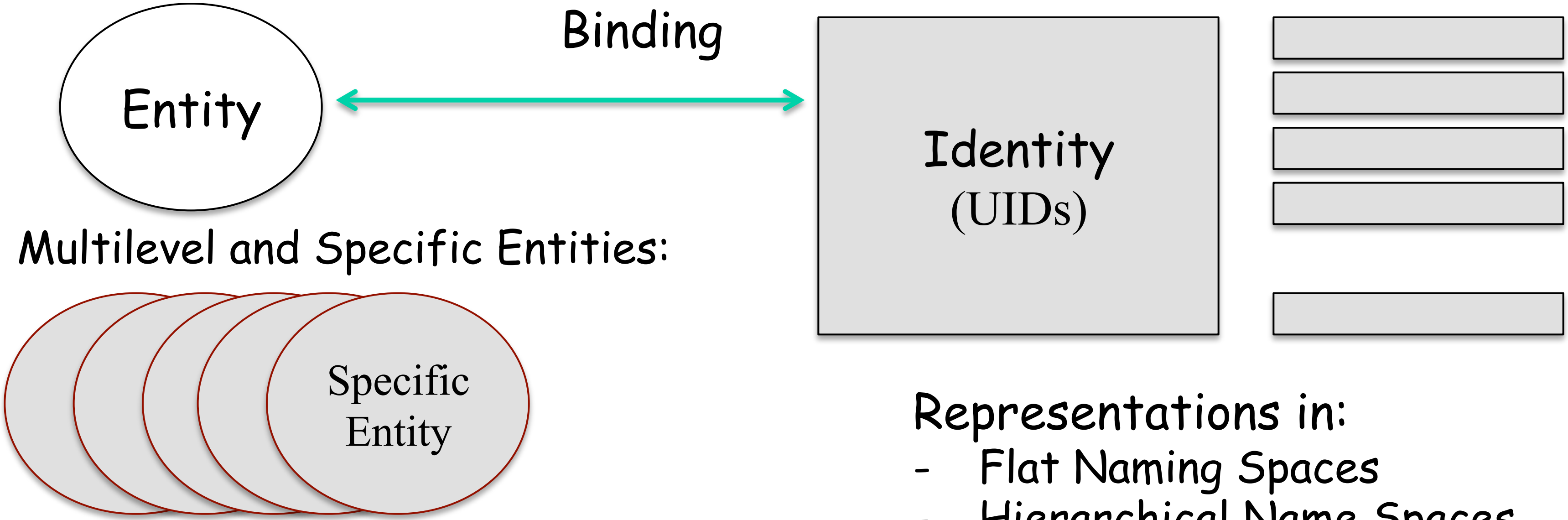
For A3 and A8, usually adopts the COMP-128 symmetric algorithm series (GSM Cons. Recommendations)

Outline

- Authentication
- Common Authentication Protocols and Services
- Authentication approach levels: Multilayer Authentication
- Authentication Systems and Frameworks
- Federated Identity Management
- SSO with OpenID
- SSO with SAML

Generic Identities

- A Digital Identity representing a named entity, represented as a set of attributes related to the entity
  - Different Types of Attributes: Pure/Specific or Contextual



Multilevel and Specific Entities:

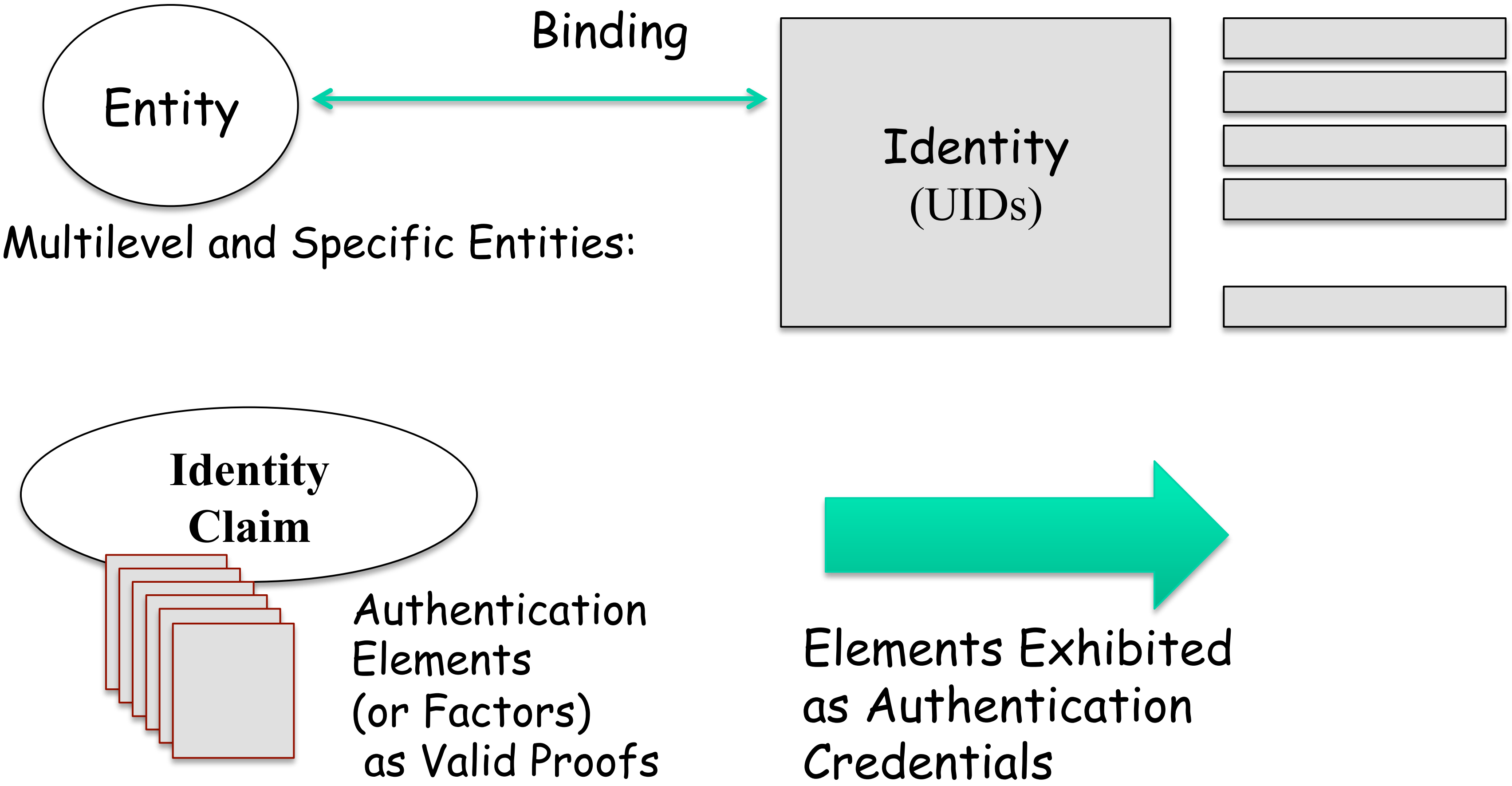
Persons, Humans
Services
Resources
Computers, Devices, Machines
SW Components, Specific FW
Chips

Representations in:

- Flat Naming Spaces
- Hierarchical Name Spaces
  - Ex.
    - X500
    - DNS
    - OIDs
    - URIs (URNs or URLs)

Identity Claim < > Authentication

- A Digital Identity representing a named entity can be represented as a set of attributes related to the entity



Generic concerns (1)

- **Trust and proof (as evidence) strength**
  - How good (how strong) is the authentication proof for a specific requirement ?
  - What is the confidence level on the exhibited proof ?
  - How difficult is it to subvert the authentication proof / evidence ?
- **Proof Secrecy**
  - Non-disclosure guarantees of credentials used by legitimate entities
    - Note: Also relevant for confidentiality of possible private attributes related to authentication proofs that may have to be protected
      - Examples ? Discussion

Generic Concerns (2)

- **Robustness**
  - Attacks to protocols (and their message exchanges)
  - Mitigation and Resistance against possible interactive DoS Attacks
  - "Off-Line Attacks" against weaknesses of critical authentication data
    - Ex., Off-Line Dictionary Attacks
- **Simplicity and Usability**
  - Simplicity enough for usability to avoid "dangerous simplifications or practices to subvert the correct operation"
- **Auditability**
- **Resistance against misuse**
  - Particularly relevant when humans (errors, unawareness, misuse practices) are present

Generic Concerns (3)

- **Temporal or Contextual Validity**
  - Continuous Authentication (as long as the interaction takes place)
  - Session-Oriented Authentication
    - "Session" contextually related to different validation criteria
      - Number of Operations
      - Criticism Level of Operations
      - Time-based validity : < From, to > ; < From, duration>
      - Connection-Oriented Mapping (ex., TCP Connections)
- **Interaction Model**
  - Peer-to-Peer Authentication
    - End-to-End Authentication
    - Point-to-Point Authentication
  - Group-Oriented Authentication
- **Directionality**
  - Unilateral Authentication
  - Bidirectional or Mutual Authentication

Mutual Authentication and Fairness

- Mutual Authentication Fairness Guarantees:
- A complementary concern that can be considered: guarantees of correct termination on mutual authentication protocols

- Fair mutual authentication vs. non-repudiation guarantees
  - Interesting ? … Why ?
  - Possible Solutions ? Discussion …

Elementary Authentication Approaches

- One-Shot Credentials vs. Challenge/Response Approaches
- Direct Authentication: Interchange of Authentication Credentials and verdict only involving the principals as direct peers
- Intermediated Authentication: Interchange of Authentication Credentials and verdict decision via a Third-Party Trusted Entity
  - Intermediation can involve partially or totally those third-party entities
  - Ex: SSO (Single Sign On) as a form of Intermediated Authentication
- Delegated Authentication: Authentication credentials presented by an entity that can represent the authenticated principal
  - Also considered as a form of indirect authentication

Authentication as a Multi-Level Concern

- Authentication of interacting entities (principals) at different levels of approach
- Use of proper authentication elements (or factors) for authentication proofs of claimed digital identities
  - Humans, Persons (User-Level)
  - Services, Servers (DNS FQNs, X509 Certificates
  - Networks
  - Machines, Devices
    - IP Addresses
    - MAC-Level Addresses
    - <IP, Port> Processes
    - OS-Level Authentication and/or Firmware-Authentication
      - Can include Attestation Proofs for BIOS + Boot Loading + SW/FW Components
    - Hardware-Based Authentication
      - Can include roe example Attestation Proofs for HW Components (CPUs, TPMs, etc)

UIDs and Authentication Elements

- Examples:
  - **Unique Ids, ex**: Email Addresses, DNS/FQ Names, Email-Delivery/ Route Endpoints, IP Addresses, <IP,Port> endpoints, MAC-Addresses, WEB/URLs, Software Attestation Ids, HW-based Identifiers, …
  - **Possible related attributes, ex:**
    - Time, Location, Interaction Context, …
  - **Authentication factors or elements**: Peer-Authenticable Digital Signatures, Message-Authentication, Authentication Cookies, Authenticable Challenge/Response Nonces, Authenticated Location, Authenticated Time
- Authentication factors or elements: used as proofs of claimed identifiers (exhibited/verified) in Authentication Protocols performed by computing systems (endpoints)

User UIDs and Authentication Elements

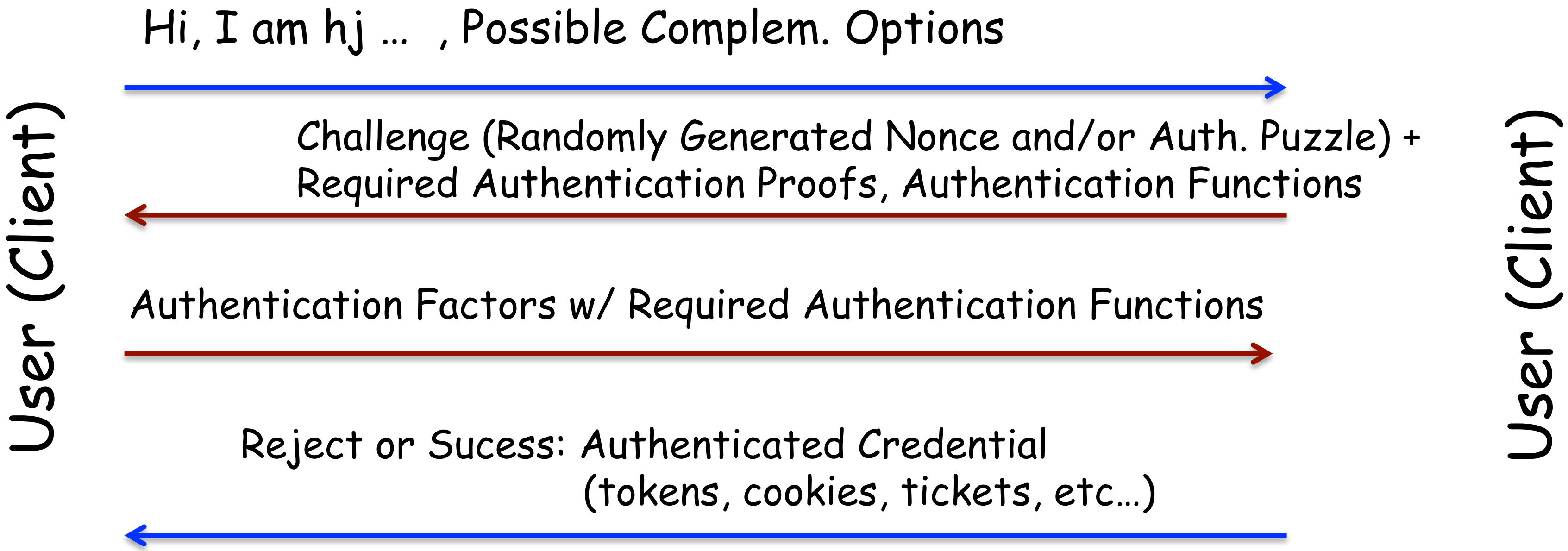
- … as proofs of claimed user identifiers (and related attributes):
  - **Unique Ids, ex**: hj, uid, hj@fct.unl.pt, Citizen Card Nr, Credit Card Nr, Bank Contract/Acc Nr, SSN, VAT Nr, DL Nr, Insurance Policy Contract Nr … etc.
  - **Possible related attributes, ex:**
    - hair, nationality, color of eyes, height, age, date-of-birth, … etc.
    - time, validity-time, location, authentication context, health condition, … etc.
  - **Authentication factors or elements**: PWD, Implicit or Explicit Secrets, CC card, DL Card, Fingerprint, Iris Structure, Voice, Handwriting Sig., Keyboard Writing Profile, Explicit or implicit cognitive elements ...
- Authentication factors or elements: used as proofs of claimed digital identities as user-level identifiers

Multi-Layered Authentication

- When different levels of authentication are orthogonally considered in a possible ***secure layering strategy***
- Remembering the layering security principle:
  - **Layering** refers to the use of **multiple, overlapping protection approaches addressing the people, technology, and operational aspects of information systems**. By using multiple, overlapping protection approaches, the failure or circumvention of any individual protection approach will not leave the system unprotected. We will see throughout this book that a layering approach is often used to provide multiple barriers between an adversary and protected information or services. This technique is often referred to as *defense in depth*.
- Can also consider the strategy as a "multi-level-stack authentication", with possible layering combinations

Example of multi-layering

- Multi-layered Authentication Enforcement
- Ex., of a Principal w/ a multilayered authentication profile:
  - hj, w/ Email hj@fct.unl.pt, VAT Nr XXXXXXX, CC NNNNNNN, at Deice IMEI Nr, IP Address (or DNS Name) port TCP, w/ this X509v3 certificate, using a WLAN NIC Card MAC Address, at location X, Age, Brown Eyes, 1m80cm height, …



User (Client)

Hi, I am hj … , Possible Complem. Options

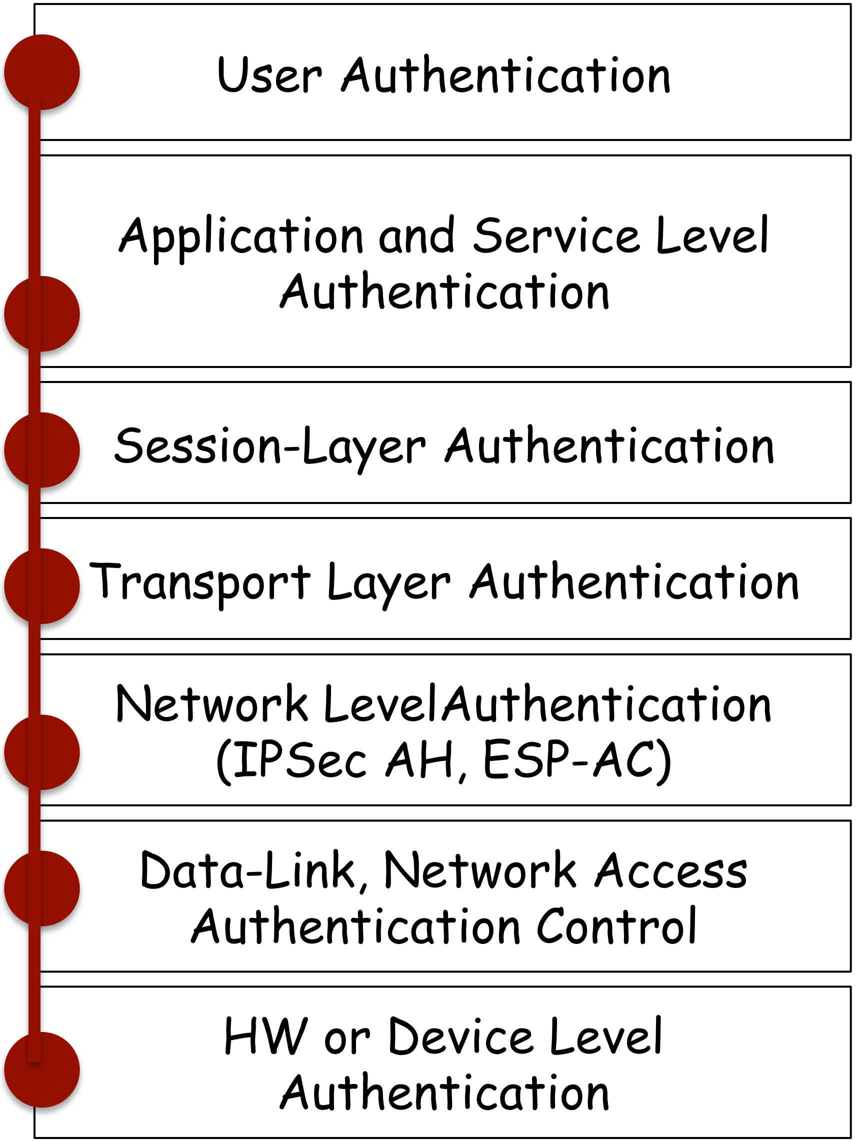
Challenge (Randomly Generated Nonce and/or Auth. Puzzle) + Required Authentication Proofs, Authentication Functions

Authentication Factors w/ Required Authentication Functions

Reject or Sucess: Authenticated Credential (tokens, cookies, tickets, etc…)

User (Client)

Multi-Layering Authentication Factors



User Authentication

Application and Service Level Authentication

Session-Layer Authentication

Transport Layer Authentication

Network LevelAuthentication (IPSec AH, ESP-AC)

Data-Link, Network Access Authentication Control

HW or Device Level Authentication

User Authentication Factors and
User Authentication Methods:
Ex: PWDs, SmartCards, Tokeks, Biometry

Kerberos, Email Authentication (ex., PGP, S/MIME), X509 Authentication, DKIM, Secure POP(3/4), Secure IMAP
HTTPS Authentication, SSH-based Applications, RADIUS

TLS, DTLS Authentication, SSH Authentication
WTLS Authentication

IPSec Authenticated Protocols
(IPSec AH, ESP-AC)

802.11i (802.11i RSN – Robust Secure Network) Authentication,
WEP, WPA, 802.1x,

Ex., IMEI Device Nr, TPM Public Key

Outline

- Authentication
- Common Authentication Protocols and Services
- Authentication approach levels: Multilayer Authentication
- Authentication Systems and Frameworks
- Federated Identity Management
- SSO with OpenID
- SSO with SAML

Authentication and Authentication Frameworks

- What can be involved in Authentication Systems ?
  - One or more unique identifiers
    - Can be managed on orthogonal Identity Management Systems (ex., Federated Identity Management)
  - One or more optional attributes (private or public)
  - Protocol
  - Service
    - Orthogonal Services (ex., SSO, Kerberos, OAuth, …)
    - Specific Authentication Components
  - Interaction model
  - Authentication elements and their

    (means or factors for the verification proof)

    The Proof can be based in one or more authentication elements (proof with multiple authentication factors - or multifactor authentication)

Standardization and Representation Issues involved in Authentication Frameworks

Authentication Services and Protocols (1)

See previous topics (class lecture and worl-assignments) on authentication protocols and services:

- Specific Authentication Protocolos:
  - Authentication and Key Distribution Protocol Protocols (A-KDP) based on studied A-KDP Reference Models
    - Can use symmetric and Asymmetric Encryption, using KDCs or PKCs
  - Base Challenge/Response Authentication Protocols
  - Password-Based Authentication and Authentication based on Password-Based Encryption Methods
  - Simple and common authentication protocols and services:
    - PPP PAP, CHAP (RFC 1334 /1992, RFC 1994 /1996);
    - MS-CHAP (RFC 2433 /1998, RFC 2759 /2000
    - RADIUS

Authentication Services and Protocols

- Orthogonal Authentication Systems
  - Kerberos AUthentication
  - X509 Authentication
  - OAuth Providers (see OAuth 2, https://oauth.net)
  - OpenID and OpenID Providers (see https://openid.net)
  - SAML (Security Assertion Markup Language) Providers
    - See SAML 2 (https://en.wikipedia.org/wiki/SAML_2.0)
    - See also https://developers.onelogin.com/saml

Orthogonal Authentication Services have an approach for use as SSO (Single Sign On) Systems for Authentication Intermediation

SSO Systems

- Usable (orthogonally) by different (independent) software systems with possible independent auditing of Single Sign On / Single Sign Off functions
- SSO systems must internally store the credentials used for initial authentication and translate them to the credentials required for the different mechanisms involved (as required by the service providers).

Benefits:

- Mitigate risk for access to 3rd-party sites (user passwords not stored or managed externally)
- Combines techniques to ensure that principals (mainly targeted to users) do not have to actively enter their credentials more than once / Reduce Password Exposition/Fatigue/Weaknesses
- Reduce time spent re-entering passwords for the same identity
- Reduce IT costs / Specialization and Auditing of Authentication Processes

SSO Systems and Criticism

- Limitations in addressing specific or different levels of authentication elements for secure access control

- Can increase the negative impact in case the credentials are available for incorrect and misused ends
  - SSO requires an increased focus on the protection of users' credentials, and should ideally be combined with externally verifiable strong authentication methods

- SSO systems are highly critical as possible single failure points;
  - A loss of availability can result in DoS to all systems unified under the SSO scrutiny:
  - Serious damage if compromised
  - Require fault and intrusion tolerance mechanisms and session failover capabilities for recovery in order to maintain the system operation

"Outsourced" SSO Systems and Criticism

- "Universal & Cheap" Outsourcing SSO Approaches: Authentication Delegation on Social Networking Authentication Services or other "Third-Party Outsourced" SSO systems
  - EX: OpenID, Google, Facebook, Janrain, Freelancer, FarmVille, Sears.com … and other Oauth v2 SSOs
  - Is it a way to go ? Discussion

- Also may render third party websites unusable within libraries, schools, or workplaces that block social media sites for productivity reasons

- It can also cause difficulties in countries with active censorship regimes
  - Sometimes the third party website may not be actively censored, but is effectively blocked if a user's social login is blocke

Outline

- Authentication
- Common Authentication Protocols and Services
- Authentication approach levels: Multilayer Authentication
- Authentication Systems and Frameworks
- ➡ Federated Identity Management
- SSO with OpenID
- SSO with SAML

FIM - Federated Identity Management

- Relatively new concept dealing with the use of a common identity management scheme across multiple domains (ex., organizations or enterprises) and numerous applications to support thousands, even millions, of users.
  - **This is the notion of Federation**
  - A process where authentication and permission will be passed on from one system to another— usually across multiple domains, thereby reducing the number of authentications needed by the user.
- The means of linking a person's digital identity and attributes stored across multiple distinct identity management systems (implementing different trust identification domains)
- Relates with Identification Management for SSO Authentication Models

Identity Management Principles for FIM

- The focus of identity management in FIM is defining an identity for users (human or human-driven process)

- Association of attributes with the identity, and enforcing a means by which a user can verify identity.

- The central concept of identity management in the FIM approach is to target sign-on (SSO). enabling a user to access all network resources after a single authentication.

FIM vs. Base Identity Management Elements

- **Authentication**
- **Authorization:** Granting access to specific services and/or resources based on the authentication.
- **Accounting:** A process for logging access and authorization.
- **Provisioning:** The enrollment of users in the system.
- **Workflow automation:** Movement of data in a business process.
- **Delegated administration:** The use of role-based access control to grant permissions.
- **Password synchronization:** Creating a process for single sign-on (SSO) or reduced sign-on (RSO): enables a user to access all network resources after a single authentication. RSO may involve multiple sign-ons but requires less user effort than if each resource and service maintained its own authentication facility.
- **Self-service password reset:** Enables the user to modify pwds

Kerberos as a FIM and SSO System

- Note that Kerberos contains a number of the elements of an identity management system and SSO approach
  - Discussion: What are the key-elements that can be associated to Kerberos (V5) as a FIM and SSO Authentication System ?

Identity Federation

- An extension of identity management to multiple security domains.
  - Such domains include autonomous internal business units, external business partners, and other third-party applications and services.

- Goal: to provide the sharing of digital identities so that a user can be authenticated a single time and then access applications and resources across multiple domains.
  - If domains are relatively autonomous or independent, no centralized control is possible.
  - Rather, the cooperating organizations must form a federation based on agreed standards and mutual levels of trust to securely share digital identities.

Identity Federation Benefits

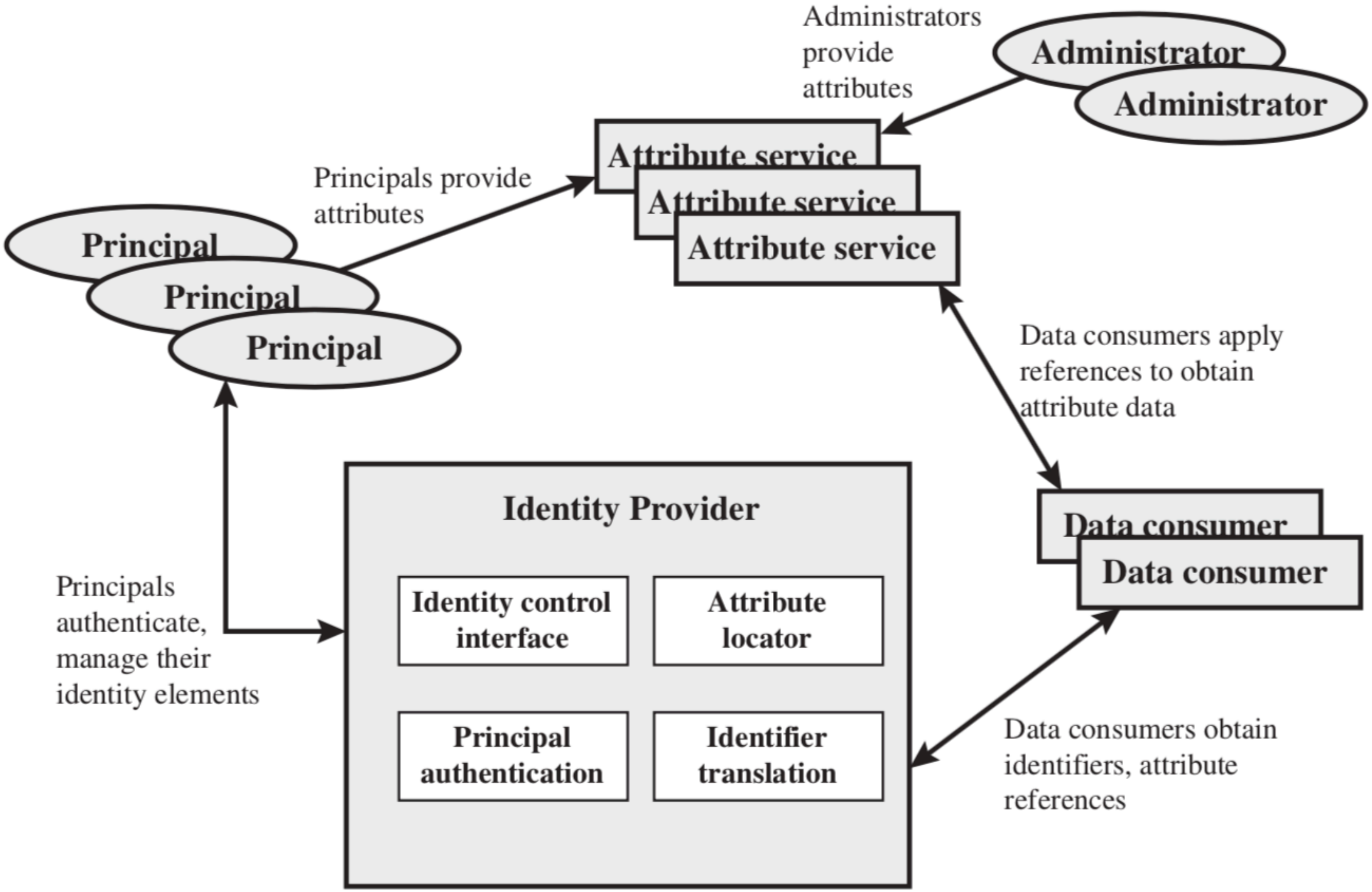
- Promotes agreements, standards, and technologies enabling the portability of identities, identity attributes, and entitlements across multiple enterprises

- When multiple organizations implement interoperable federated identity schemes, an employee in one organization can use a single sign-on to access services across the federation with trust relationships associated with the identity.

- For example, an employee may log onto her corporate intranet and be authenticated to perform authorized functions and access authorized services on that intranet.

  - The employee could then access their health benefits from an outside health-care provider without having to reauthenticate.

Generic Identity Management Architecture


Administrators provide attributes
Administrator
Administrator
Principals provide attributes
Attribute service
Attribute service
Attribute service
Principal
Principal
Principal
Data consumers apply references to obtain attribute data
Identity Provider
Identity control interface
Attribute locator
Principal authentication
Identifier translation
Data consumer
Data consumer
Principals authenticate, manage their identity elements
Data consumers obtain identifiers, attribute references

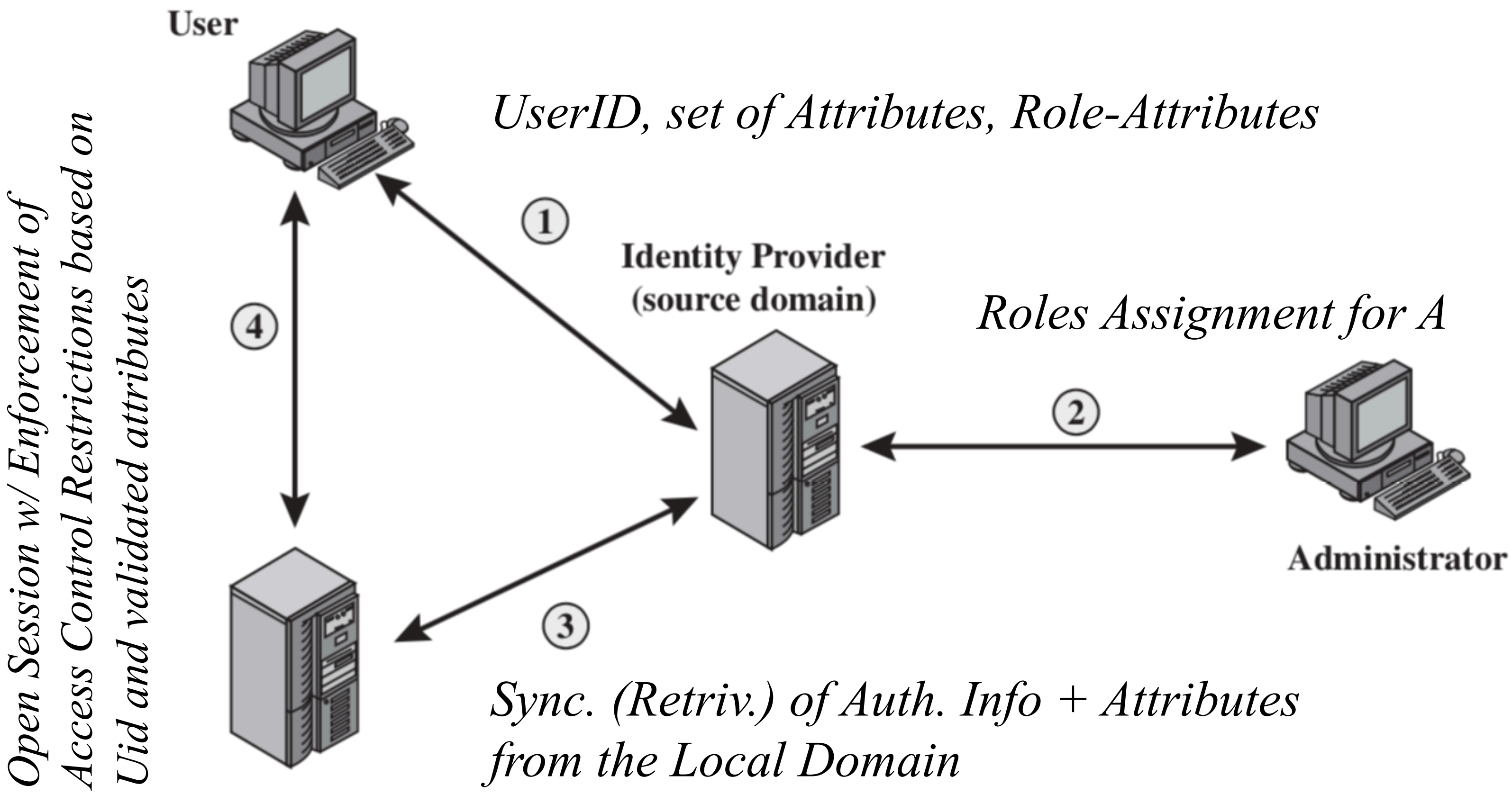
FIM, SSO and Extensibility Capabilities (1)

- Standardized means of representing extensible attributes.
  - Increasingly, digital identities incorporate attributes other than simply an identifier and authentication information (such as passwords and biometric information).
  - Examples: account numbers, organizational roles, physical location, and file ownership.
  - A user may have multiple identifiers; for example, each identifier may be associated with a unique role with its own access permissions.

FIM, SSO and Extensibility Capabilities (2)

- Identity mapping and flexibility
  - Different security domains may represent identities and attributes differently.
    - Furthermore, the amount of information associated with an individual in one domain may be more than is necessary in another domain.
  - The federated identity management protocols map identities and attributes of a user in one domain to the requirements of another domain.

FIM Operation


User
UserID, set of Attributes, Role-Attributes
1
Identity Provider
(source domain)
Roles Assignment for A
2
Administrator
4
3
Sync. (Retriv.) of Auth. Info + Attributes
from the Local Domain
Open Session w/ Enforcement of
Access Control Restrictions based on
Uid and validated attributes

Service Provider
In the destination
Domain

FIM Identity Providers vs. Service Provider

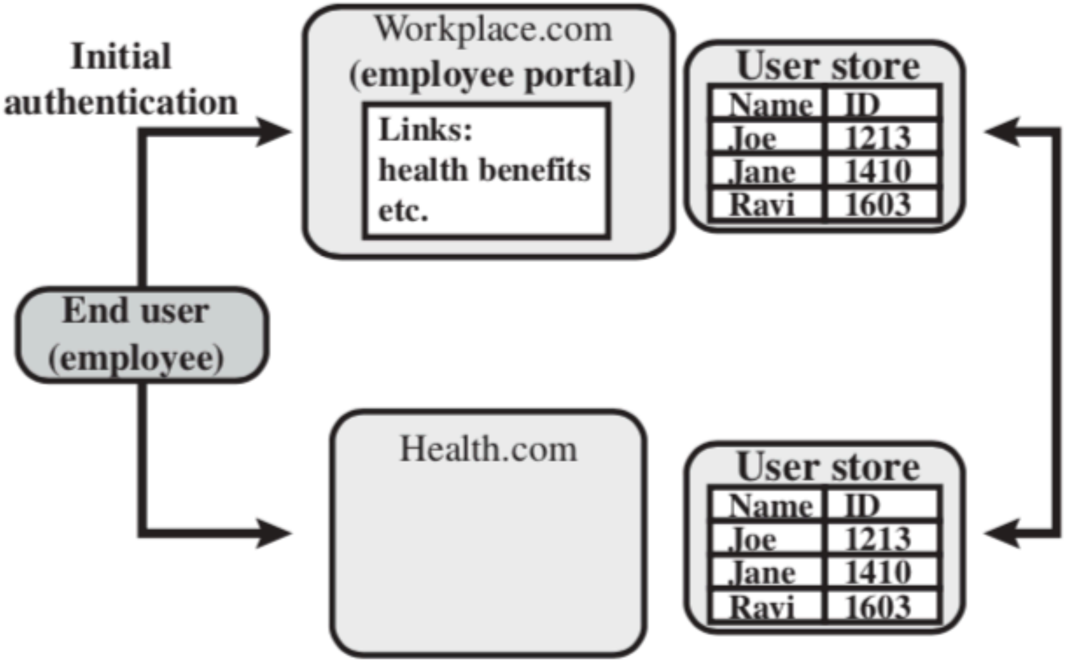
- FIP IPs: acquires attribute information through dialogue, enrolment and protocol exchanges with users and administrators.
  - Ex: user needs to provide a shipping address each time an order is placed at a new Web merchant, and this information needs to be revised when the user moves.

Then … **Identity management enables the user to provide this information once**, so that it is maintained in a single place and released to data consumers in accordance with authorization and privacy policies.

FIM Identity Providers vs. Service Provider

- FIP SPs: are entities that obtain and employ data maintained and provided by identity providers, often to support authorization decisions and to collect audit information.
  - For example, a database server or file server is a data consumer that needs a client's credentials so as to know what access to provide to that client.
  - The power of the FIM approach is that **the service provider can be in a different domain** (e.g., a vendor or supplier network) and not in the user identification domain

FIM Scenarios: Account-Linking



Initial authentication

Workplace.com (employee portal)

Links:
health benefits
etc.

User store

Name	ID
Joe	1213
Jane	1410
Ravi	1603

End user (employee)

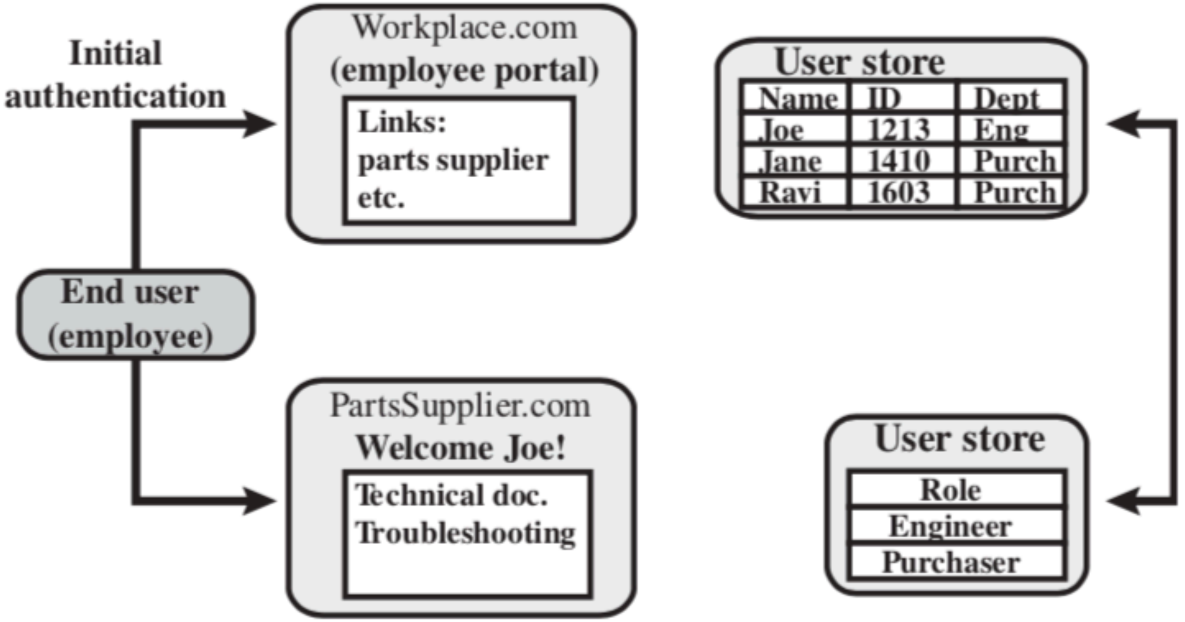
Health.com

User store

Name	ID
Joe	1213
Jane	1410
Ravi	1603

(a) Federation based on account linking

FIM Scenarios: Role-Based Federation



User store

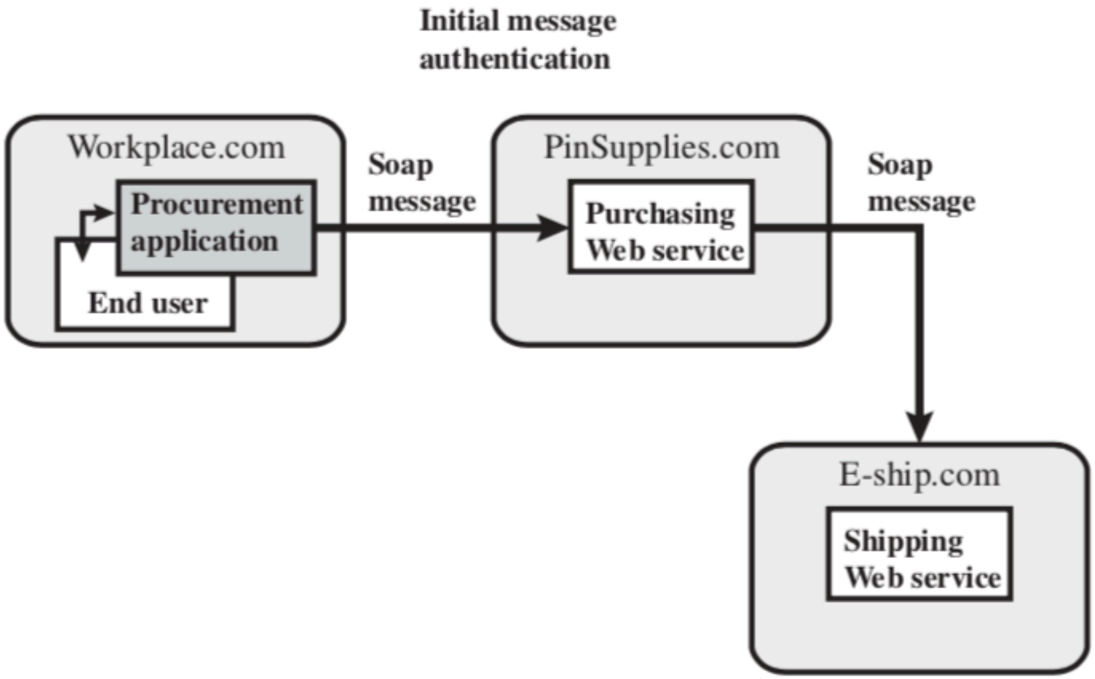
Name	ID	Dept
Joe	1213	Eng
Jane	1410	Purch
Ravi	1603	Purch

User store

Role
Engineer
Purchaser

(b) Federation based on roles

FIM Scenarios: Chained Web Services



(b) Chained Web services

Enabling Standards for FIM

- FIM uses a number of standards as building blocks for secure identity exchange across different domains or heterogeneous systems.
- In essence, organizations issue some form of "security tickets or credentials" for their users that can be processed by cooperating partners.
- FIM enabling standards are thus concerned with:
  - Defining these "tickets" in terms of content and format
  - Providing protocols for exchanging tickets, and performing a number of management tasks (including configuring systems to perform attribute transfers and identity mapping and performing logging and auditing functions).

  - FIM Open Standardization EXAMPLES:
    OpenID (Effort from the OpenID Consortium)
    SAML (Effort from the OASIS Consortium)

Outline

- Authentication
- Common Authentication Protocols and Services
- Authentication approach levels: Multilayer Authentication
- Authentication Systems and Frameworks
- Federated Identity Management
- SSO with OpenID
- SSO with SAML

OpenID

- OpenID is an Open Standard for a Decentralized Authentication Protocol (promoted by the OpenID Foundation):
  - https://en.wikipedia.org/wiki/OpenID#OpenID_Foundation

  Users create accounts on selected OpenID Authentication Providers

  Then, users can logon (on different web services accepting OpenID Authentication)

  - OpenID provides a framework for the communication between the identity provider and the OpenID acceptor (relying party)

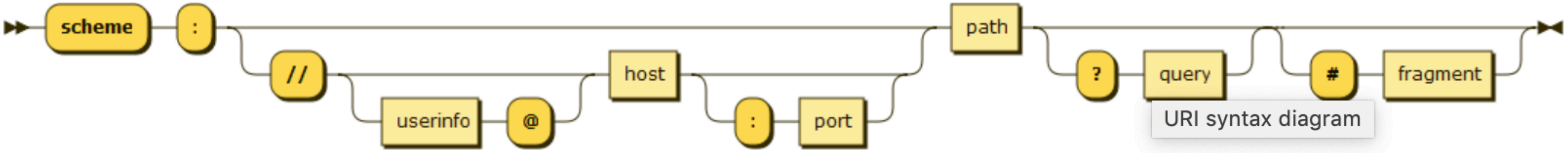
- OpenID identifiers have a form of URIs (Uniform Resource Identifiers)

OpenID: URIs (as in RFC 3986, Jan/05)

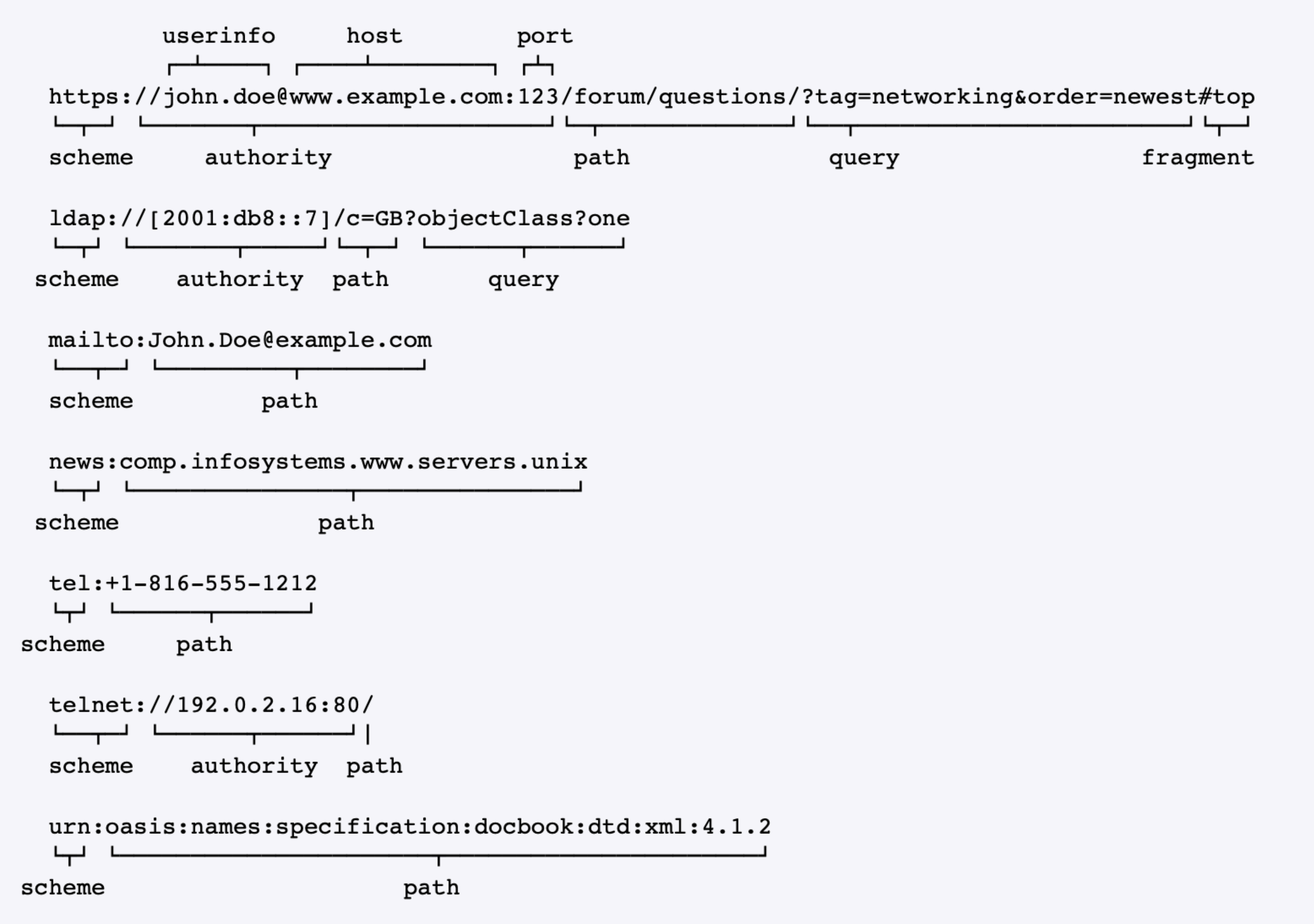
Strings of chars that unambiguously identifies a particular principal (resource in the URI terminology)
URIs can have a form of URLs (Uniform Resource Locators) or URNs (Unifirm Resource Names)

Represents a hierarchical structure defined by:
<scheme>:<path>

Ex., URN: *urn:isbn:0-486-27557-4*
URL: mailto:hj@fct.unl.pt



OpenID: URIs



```
          userinfo       host      port
          ┌──┴───┐ ┌──────┴──────┐ ┌┴┐
  https://john.doe@www.example.com:123/forum/questions/?tag=networking&order=newest#top
  └─┬─┘   └───────────┬──────────────┘└───────┬───────┘ └───────────┬─────────────┘ └┬┘
  scheme          authority                  path                 query           fragment

  ldap://[2001:db8::7]/c=GB?objectClass?one
  └┬─┘   └─────┬─────┘└─┬─┘ └──────┬──────┘
 scheme    authority   path      query

  mailto:John.Doe@example.com
  └─┬──┘ └────┬─────────────┘
  scheme     path

  news:comp.infosystems.www.servers.unix
  └┬─┘ └─────────────┬─────────────────┘
 scheme            path

  tel:+1-816-555-1212
  └┬┘ └──────┬──────┘
scheme     path

  telnet://192.0.2.16:80/
  └─┬──┘   └─────┬─────┘│
  scheme     authority  path

  urn:oasis:names:specification:docbook:dtd:xml:4.1.2
  └┬┘ └──────────────────────┬──────────────────────┘
scheme                    path
```

OpenID Protocol

- Defines a standard (interoperable) authentication protocol, supporting standardized (or extended) OpenID Attributes Exchange) from the OpenID identity provider to the relying party
  - Attributes: name, age, gender, …
  - Each relying party may request a different set of attributes, depending on their specific requirements

- The protocol does not rely on a central authority to authenticate a user's identity.
- Flexible, allowing for approaches ranging from common authentication factors (such as passwords) to other multi-factor user-authentication elements)

- Today the more conventional OpenID Protocol Implementation is OpenID Connect, based on OAuth 2

More on OpenID Reference

https://openid.net

Suggested video:

https://www.youtube.com/watch?time_continue=68&v=Kb56GzQ2pSk

Outline

- Authentication
- Common Authentication Protocols and Services
- Authentication approach levels: Multilayer Authentication
- Authentication Systems and Usual Authentication Protocols
- SSO with OpenID
- SSO with SAML
- Federated Identity Management

SAML: Security Assertion Markup Language

Motivation:

- Permissions management data is currently handled in mostly proprietary ways, among tightly coupled modules in a single security domain.
- Web resources and access-control policy management is loosely coupled, consisting of many security domains.
- SAML is a standard needed to govern the transfer of assertions between different domains.

What is SAML ?

SAML is an open standard for exchanging authentication and authorization data between parties, in particular, involving:

- Clients
- An Identity Provider
- A Service Provider

- SAML is an XML-based markup language to express security assertions
  - Assertions are interoperable statements that service providers use to make access-control decisions.

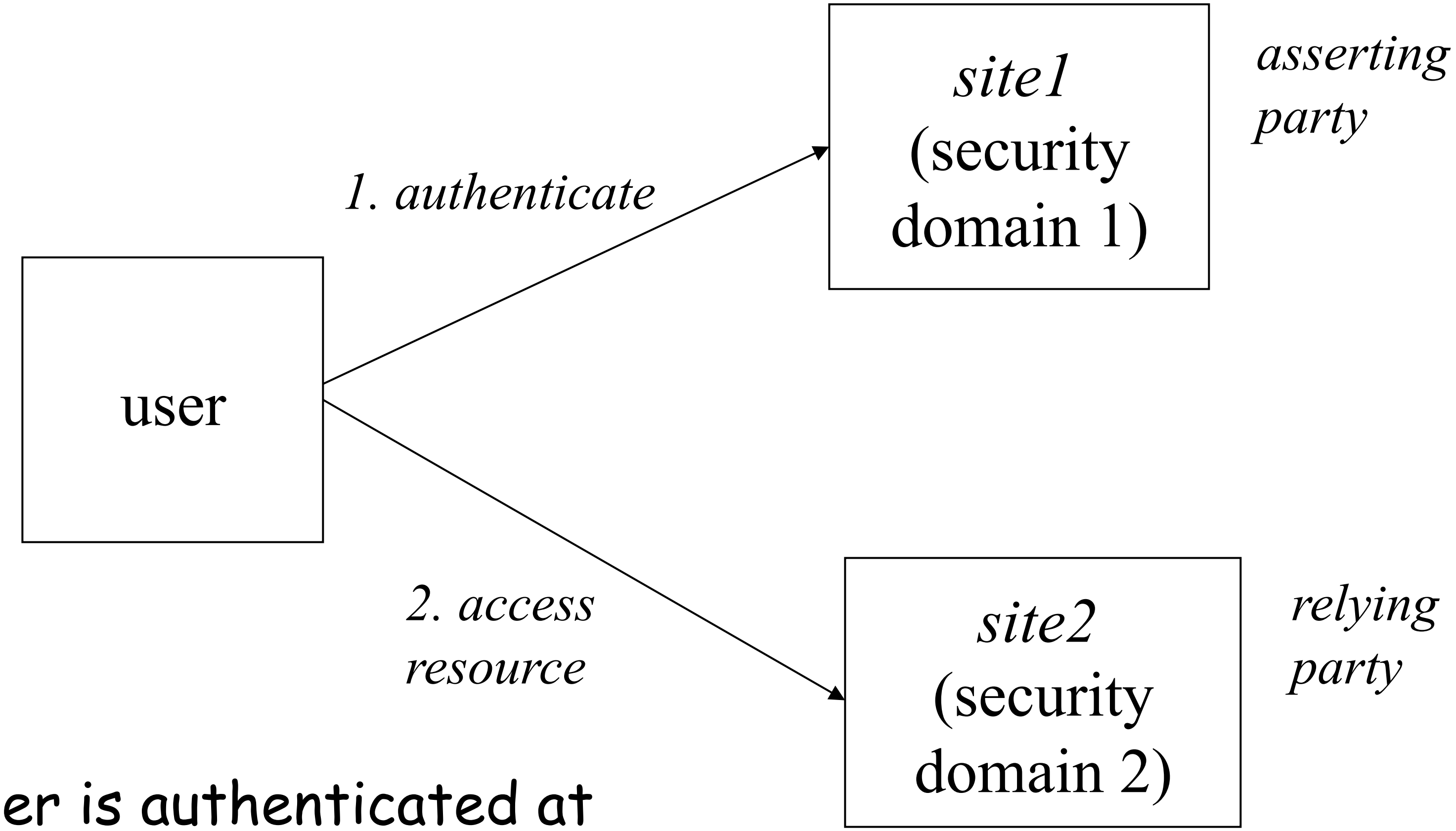
More concretely, SAML is:

- a set of XML-based protocol messages
- a set of protocol message bindings
- a set of profiles (including all of the above)

SAML Standardization by OASIS
https://www.oasis-open.org

- OASIS: **Organization for the Advancement of Structured Information Standards**
- SAML 1.0 (2002), SALM 1.1 (2003), SAML 2.0 (2005)
- The Liberty Alliance contributed its Identity Federation Framework (ID-FF) to the OASIS SSTC in September 2003
  - ID-FF 1.1 was released in April 2003
  - ID-FF 1.2 was finalized in November 2003

SAML Use Case for SSO



user

1. *authenticate*

site1 (security domain 1)

asserting party

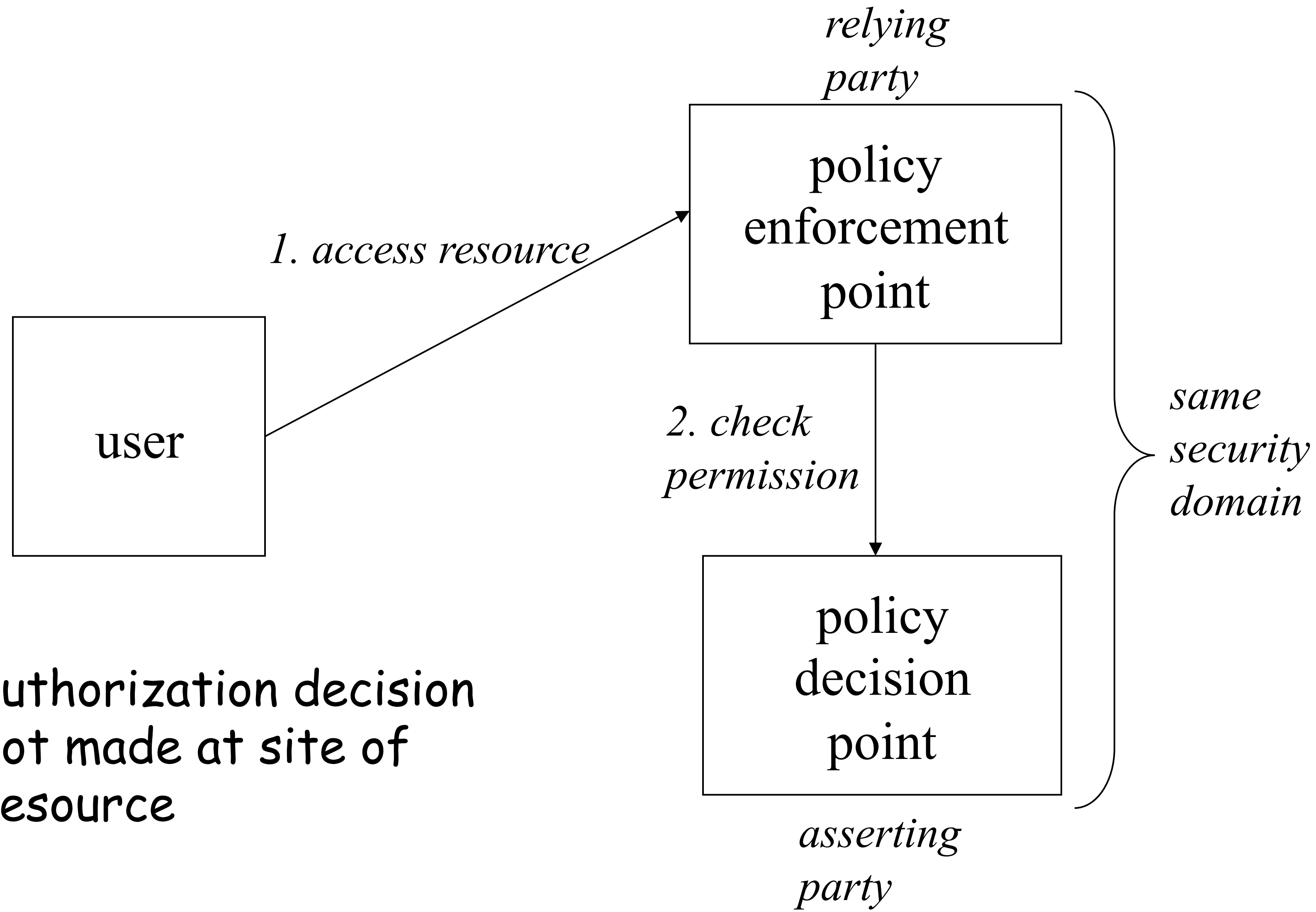
2. *access resource*

site2 (security domain 2)

relying party

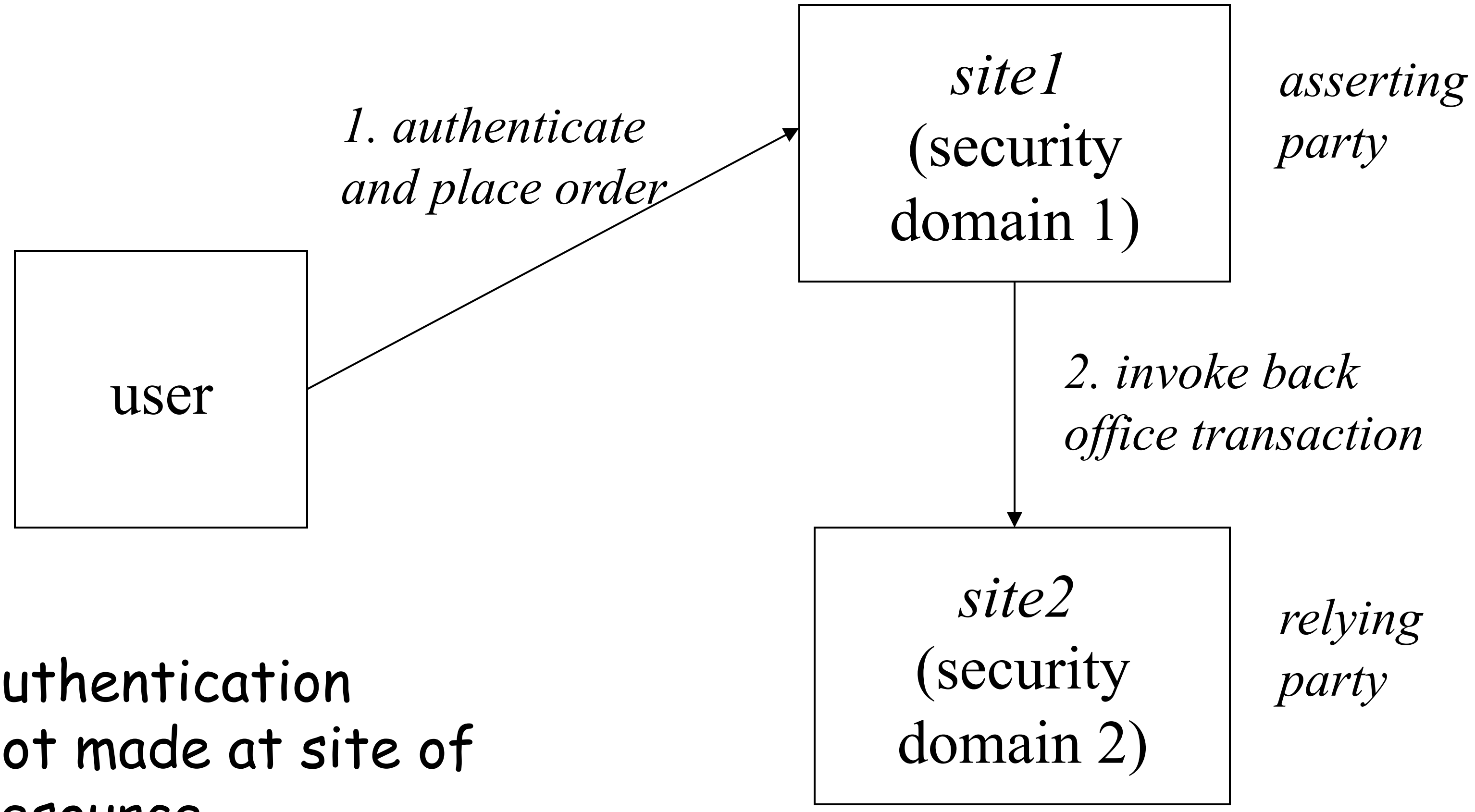
user is authenticated at *site1*; then accesses a resource at *site2*

SAML Use Case SSO for Authorization



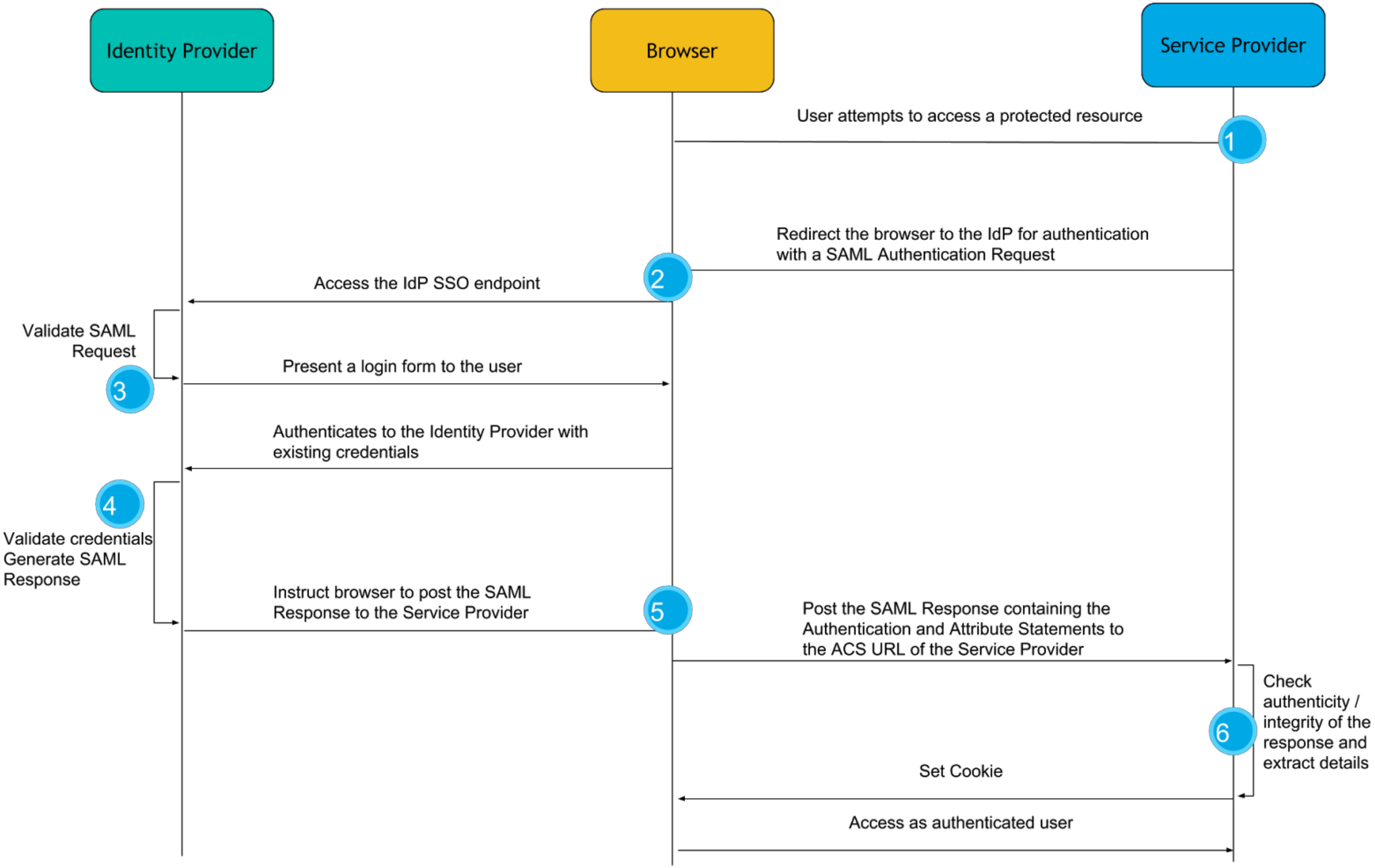
authorization decision not made at site of resource

SAML Use Case for Transactions


user
1. authenticate and place order
site1 (security domain 1)
asserting party
2. invoke back office transaction
site2 (security domain 2)
relying party

authentication not made at site of resource

SAML 2.0 Interactions


Identity Provider
Browser
Service Provider
User attempts to access a protected resource
1
Redirect the browser to the IdP for authentication with a SAML Authentication Request
2
Access the IdP SSO endpoint
Validate SAML Request
3
Present a login form to the user
Authenticates to the Identity Provider with existing credentials
4
Validate credentials Generate SAML Response
Instruct browser to post the SAML Response to the Service Provider
5
Post the SAML Response containing the Authentication and Attribute Statements to the ACS URL of the Service Provider
Check authenticity / integrity of the response and extract details
6
Set Cookie
Access as authenticated user

Why SAML is more than "simple authentication cookies" ?

- SAML is different than Cookies
  - Cookie (is a token piece signed with server's private key) that can be used for re-authentication at a particular server, but is not necessarily usable at a different server (different security domain assertion)
- Cross domain authentication requires extended support
- SAML intended as an Open Web Standard initiative to overcome interoperability problems from proprietary software solutions.

- SAML must be used in the context of a trust relationship between **asserting** and **relying parties**
  - **Example**: statement "Bill has access to resource *X*" may be of no use unless we know that Bill is at the other end of the line
  - Trust relationship is established using mechanisms such as TLS, Digital Signatures, Encryption, etc: This elements of a security framework is not part of SAML

What is an “Assertion” in SAML Terminology ?

- An Assertion is a set of statements (claims) made by a SAML authority (asserting party)

- Composed by:

  - **Authentication statement**: subject was authenticated using a particular technique at a particular time

  - **Attribute statement**: particular attribute values are associated with the subject

  - **Authorization decision statement**: subject is authorized to perform certain actions

Assertion in SAML

```
<saml:Assertion  xmlns:saml=“….”
       …version information goes here…
        AssertionID=“….”
        IssueInstant=“….”>
    <saml:Issuer> www.acompany.com </saml:Issuer>
    <ds:Signature>  … XML Signature goes here …   </ds:Signature>
    <saml:Subject>
        <saml:NameIdentifier  ….>  uid=joe  </saml:NameIdentifier>
    </saml:Subject>
    <saml:Conditions   …. />
     … SAML statements go here …
</saml:Assertion>
```

SAML authority making the claim

entity about which the claim is being made

Signatures in SAML

- A signed assertion supports
  - Assertion integrity
  - Authentication of *creator of assertion* (the SAML authority)
- A signed protocol request/response message supports
  - Message integrity
  - Authentication of *message origin* (asserting party) (might be different from creator)
- A signature is not always needed
  - Assertion might *inherit* signature of containing message
  - Assertion might be received over a secure channel whose other end was authenticated by other means
- Signature is a restricted version of XML Signature

SAML Subject

- Identifies the entity to which the assertion pertains

- Identifies confirmation method and (optionally) confirmation data

  - If the relying party performs the specified authentication method (perhaps using the data) then it can treat the entity presenting the assertion as the entity that the SAML authority associates with the name identifier

  - Example: method = public key, data = key information

Conditions

- SAML conditions are restrictions under which the assertion is to be used
  - **NotBefore** – earliest time at which assertion is valid
  - **NotOnOrAfter** – latest time at which assertion is valid
  - **AudienceRestrictionCondition** – assertion is addressed to a particular audience
  - **DoNotCacheCondition** – assertion must be used immediately
  - **ProxyRestrictionCondition** – limitation that the asserting party places on a relying party that wishes to create its own assertion based on this assertion

Authentication Statement

```
<saml:AuthenticationStatement
        AuthenticationMethod=“password”
        AuthenticationInstant=“….” />
```

- *Asserts that the enclosing assertions’ subject was* authenticated by a particular means at a particular time
  - Authentication itself is *not* part of SAML
  - Statement refers to an authentication act that took place at a prior time

Attribute Statement

```
<saml:AttributeStatement>
    <saml:Attribute  Name =“attrib”>
        <saml:AttributeValue> val </saml:AttributeValue>
    </saml:Attribute>
</saml:AttributeStatement>
```

- Asserts that the enclosing assertion’s subject is associated with attribute *attrib* with value *val.*
  - Example: the value of the attribute *Department* associated with the assertion’s subject is *Accounting*

Authorization Decision Statement

```
<saml:AuthorizationDecisionStatement  Decision=“permit”
                                      Resource=“… some URI … >
    <saml:Action> Execute </saml:Action>
</saml:AuthorizationDecisionStatement>
```

- *Asserts that the enclosing assertion's subject's request for a particular action at the specified resource has resulted in the specified decision*

SAML Based Protocols

- SAML Protocols follow a request/response pattern

- SAML specification defines protocols/messages that:
  - Request an assertion identified by unique Id
  - Request assertions containing authentication statements about the subject
  - Request assertions containing attribute statements concerning a particular attribute relating to the subject
  - Request assertions containing authorization decision statements concerning a particular resource and subject
  - Request that an authentication assertion of a particular type be created (this might involve execution of an authentication protocol)
  - Transmit protocol message by reference (artifact protocol)

SAML Profiles

- SAML defines message exchange patterns that illustrate how SAML assertions can be exchanged to achieve particular goals in a particular context
  - Involve the use of SAML protocols

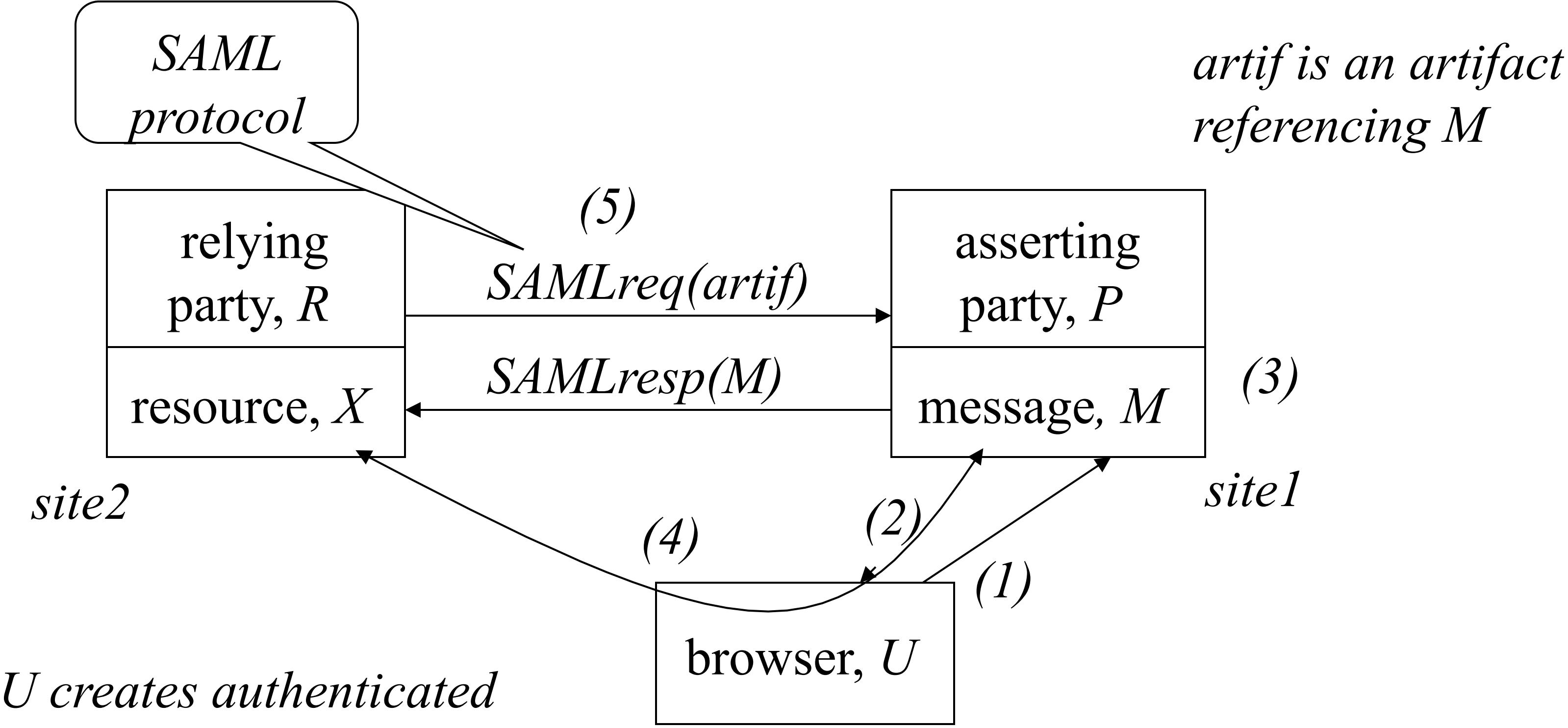
Web environment: Browser/Artifact Profile

- Browser, authenticated at *site1* (asserting party) requests access to a resource at *site2* (relying party).
  - *site1* creates a protocol message containing an authentication statement and a reference to that message called an **artifact**
  - *site2* pulls the protocol message from *site1* using the artifact

Artifact

- A SAML Artifact is nothing more than a string consisting of
  - Identity of source site (asserting party)
  - Reference to a protocol message at source site
- Use: relying party wants to retrieve assertions in a protocol message at the asserting party; supplies an artifact that identifies the message

Artifact - Pull Model



1. *U creates authenticated session with P*
2. *U requests access to X (through P).*
3. *P creates protocol msg, M, containing assertion about U, and an artifact referring to M*
4. *Access, containing artifact, is redirected from P to R through browser*
5. *R pulls M (identified by artifact) from P*

Request Message

```
<env:Body>
   <samlp:request  xmlns:samlp="…"
      RequestID="….."
      IssueInstant="…." >
      <samlp:Artifact>
         ASDFGHasdfgh….
      </samlp:Artifact>
   </samlp:Request>
</env:Body>
```

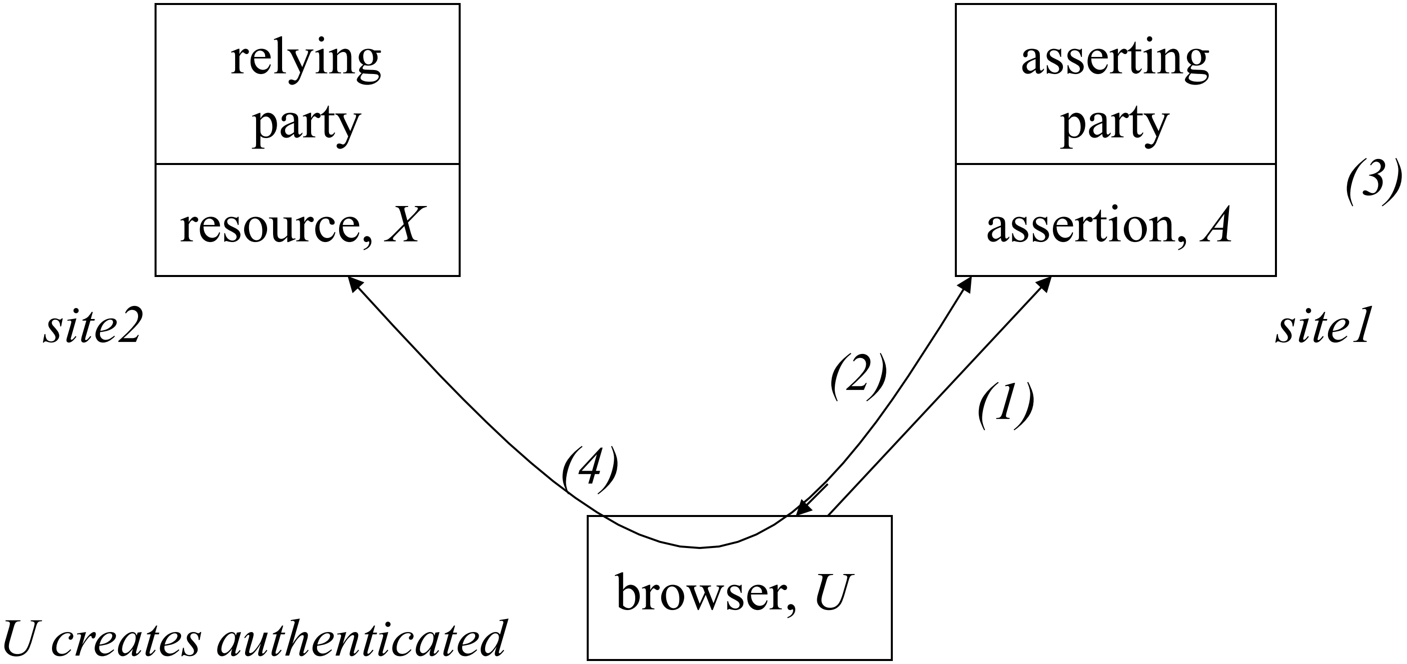
Request message (part of request/response protocol) from relying party for an assertion held by asserting party identified by artifact

Response Message

```
<env:Body>
   <samlp:Response  xmlns:samlp=“….”
      ResponseID=“….”
      InResponseTo=“….”
      IssueInstant=“….” >
      <samlp:Status>
         <samlp:StatusCode  Value=:”samlp:Success”/>
      </samlp:Status>
      … a protocol message goes here …
   </samlp:Response>
</env:Body>
```

Protocol message is returned in response message

Browser/Post Model



1. *U creates authenticated session*
2. *U accesses remote resource X through asserting party.*
3. *A asserts fact about U*
4. *Access, containing signed assertion, is redirected (pushed) through browser to relying party (signature required since assertion is routed through browser)*

SAML Security

- Message integrity and confidentiality can be achieved using TLS communication channels
- Relying party can have confidence in the assertion:
  - **Pull model**: bi-lateral authentication should be used when connection is set up between relying and asserting parties
  - **Push model**: digital signature of asserting party used on message containing assertion
  - Either way, relying party knows who asserting party is and can trust the assertion accordingly

More about SAML

See:

- https://en.wikipedia.org/wiki/Security_Assertion_Markup_Language
- SAML 2.0:
  - https://en.wikipedia.org/wiki/SAML_2.0
- Tutorial: What is SAML 2
  - https://www.onelogin.com/learn/saml
- SAML Developmen
  - https://developers.onelogin.com/saml

Revision: Suggested Readings and Study

Readings:

References in the Slides

W. Stallings, Network Security Essentials,

- Chap 4 – Key Distribution and User Authentication (Particularly Federated Based Authentication, Section 4.4)

See CLIP