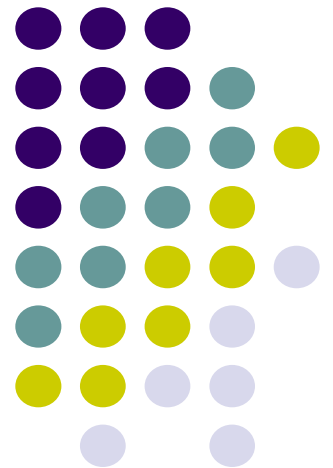


Digital Identity Management

Techniques and Policies

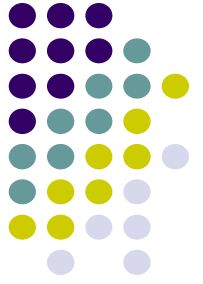
E. Bertino

CS Department and ECE School
CERIAS
Purdue University
bertino@cs.purdue.edu



Digital Identity Management

What is DI?



Digital identity (DI) can be defined as the digital representation of the information known about a specific individual or organization

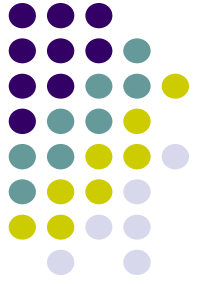
Such information is set of claims made by one subject about itself or another subject

Our definition includes both the notion of nyms – identifiers used by users to carry on interactions with systems – and *identity attributes* – properties characterizing the users

Claim: An assertion of the truth of something, typically one which is disputed or in doubt

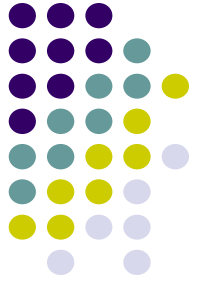
- An identifier
- Knowledge of a secret
- Personally identifying information
- Membership in a given group (e.g. people under 16)

Drivers for Dependable and Flexible DI Technology



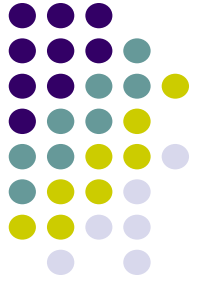
- The private sector
- The public sector
- The citizens

The goals of the VeryId project



- *To develop flexible, multiple and dependable digital identity (FMDDI) technology*
- *To study the implication of its use*
- *To develop appropriate educational vehicles to teach people its use*

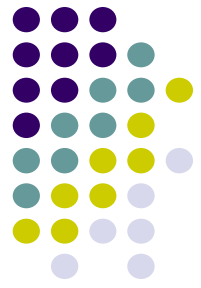
The project is funded by the USA National Science Foundation under the CyberTrust programme



Some initial results

- Protocols for the strong verification of identity attributes in federations
- Integration of biometrics
- Policies for the management of identity federations
- Authentication policies and services
- Identity provenance and quality
- Outreach activities





Identity Theft

IDENTITY THEFT is the use of personally identifying information belonging to one individual by another individual for financial or personal gain.



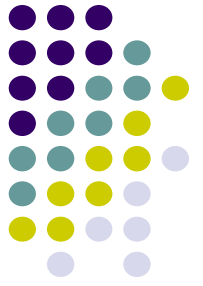
Threat of Identity Theft: Attack Vectors



Technical	Pharming, Network Sniffing, Database Attacks, Password Cracking
Physical	Dumpster Diving, Trusted Insiders, Theft and Loss
Social Engineering	Phishing, Legal Identity Sources



Main idea behind verification of identity attributes: multi-factor verification



To require additional identity information (like mother maiden name or SSN) as proof to qualify to be the owner of the identity attribute being used (like credit card number)

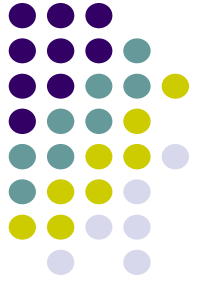
Example Real Life Scenario: Requirement for additional proofs of identity



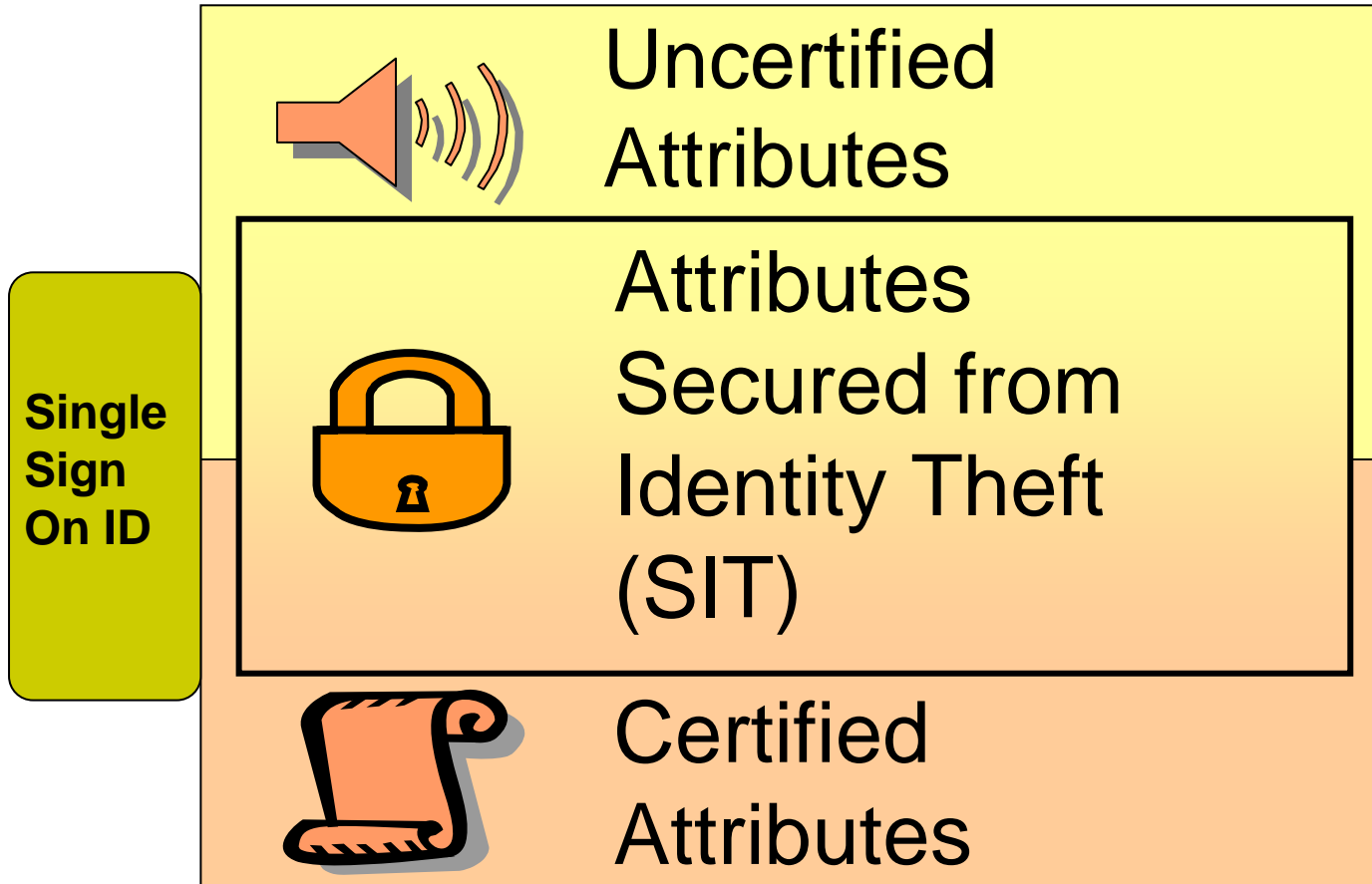


Multi-Factor without Privacy Loss

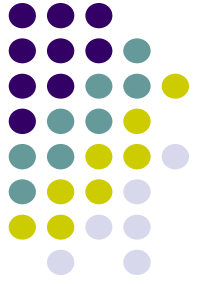
- *Zero knowledge proof (ZKP) is an interactive method to prove the possession of a secret without actually revealing it.*
- Our **aggregated ZKP scheme** is used to prove the knowledge of multiple strong identifiers efficiently and reliably without the need to provide them in clear



Attribute types



Two main phases of our solution



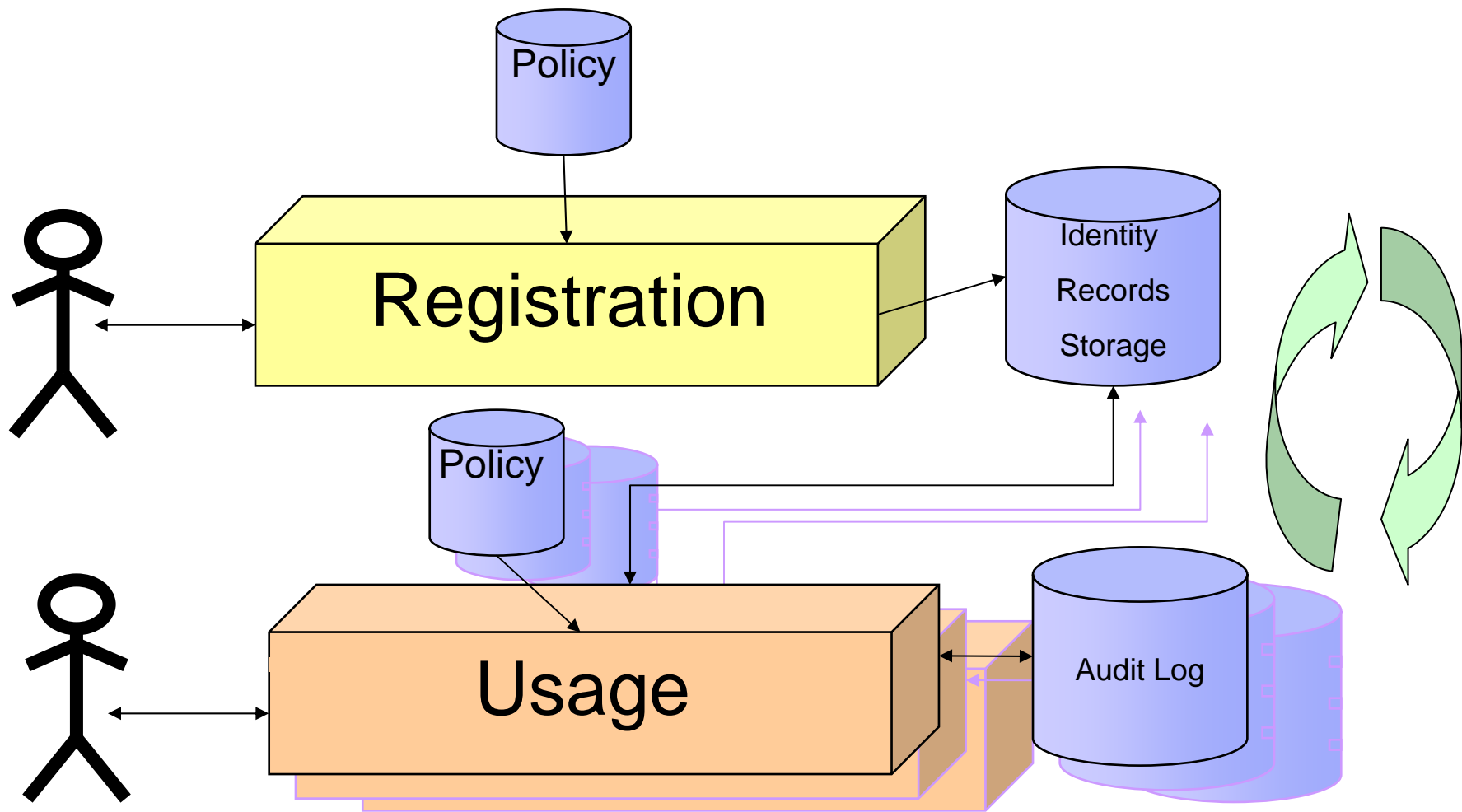
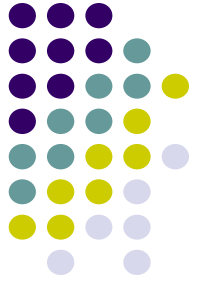
- **Enrollment or Registration**

- Here the user **commits** his strong identifiers to be used later as proofs of identity. These are the SIT attributes.

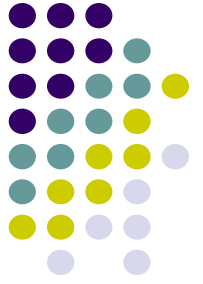
- **Usage**

- Before revealing the actual value of a SIT attribute one has to verify the commitments of other SIT attributes as *proofs of identity*.

Functional View of the System

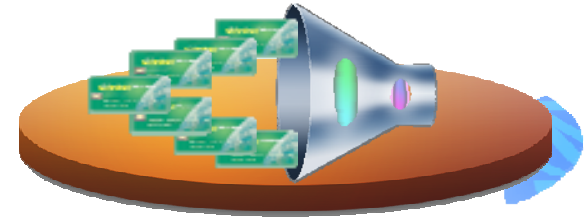


Identity Management System Entities



Relying Parties
Require identities

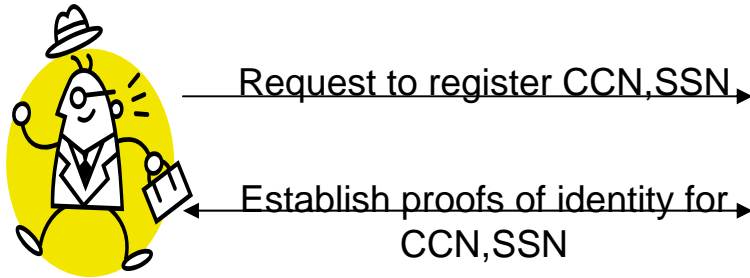
Identity Providers
Issue identities



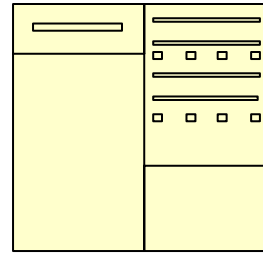
Subjects
*Individuals and other entities
about whom claims are made*



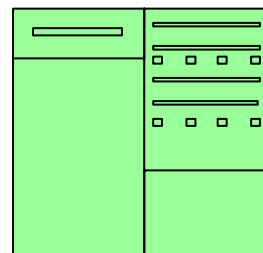
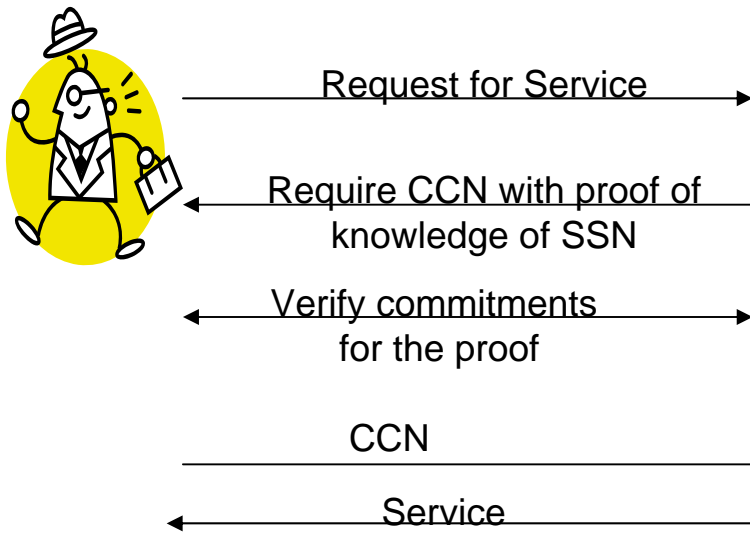
Example



Registrar or Identity Provider



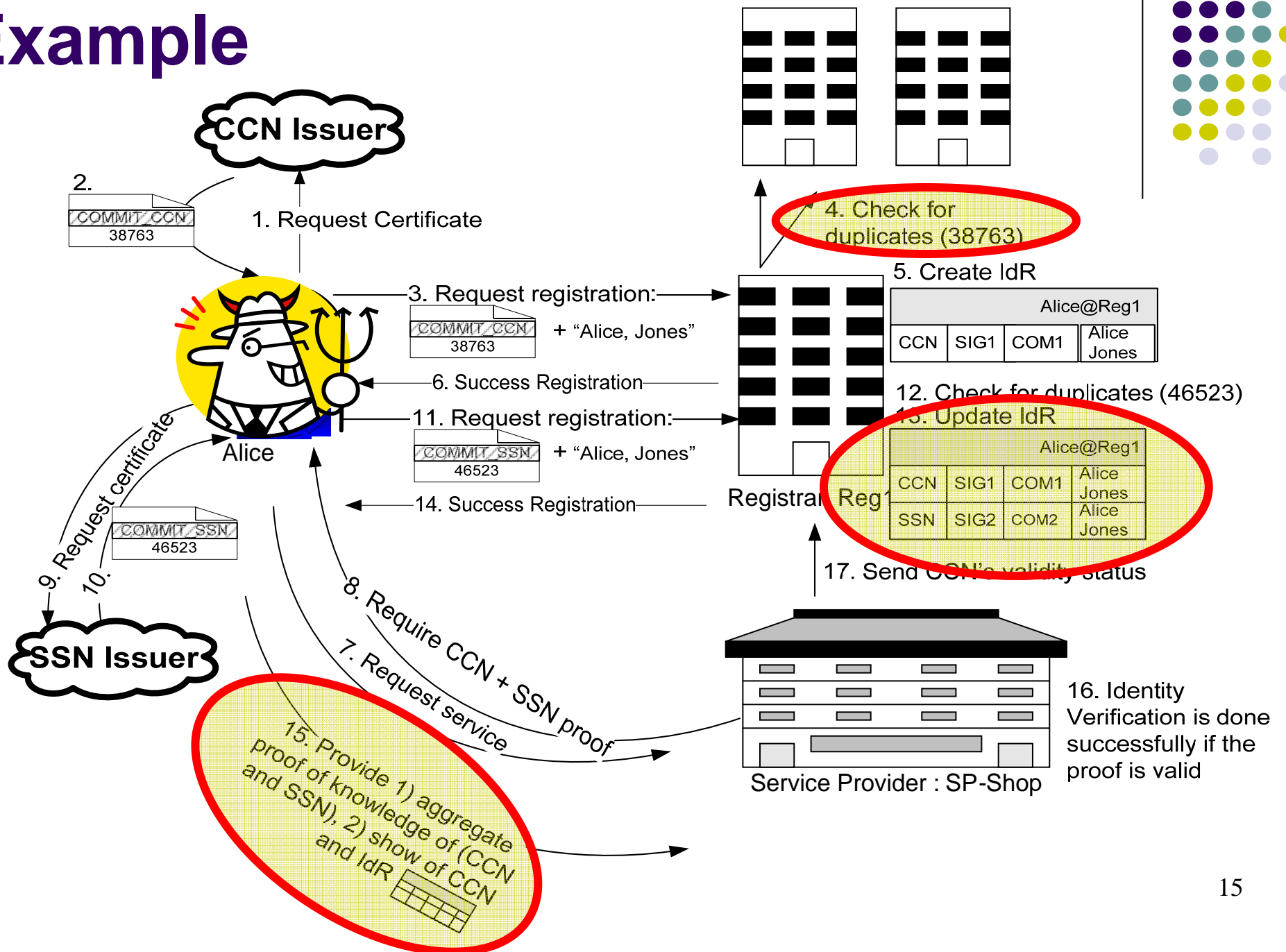
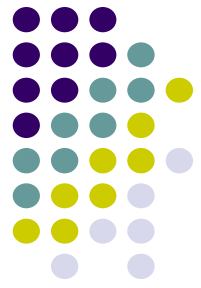
Alice@SP1		
Tag	Committed Value	Registration Procedure
SSN _{tag}	C1	In Person
CCN _{tag}	C2	Online

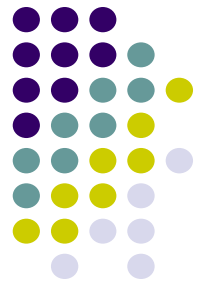


Service Provider

Registration Phase Usage Phase

Example





Proving aggregated signature on committed values

To prove the knowledge of multiple identifiers.

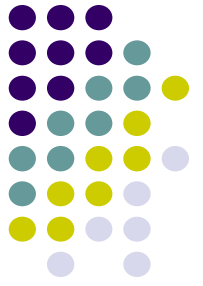
- (1) *Principal's aggregation.* Let $\sigma_1, \sigma_2, \dots, \sigma_t$ be the signatures corresponding to the tags in ψ_{proof} . The principal aggregates the signatures into $\sigma = \prod_{i=1}^t \sigma_i$, where σ_i is the signature of committed value $M_i = g_1^{m_i} h_1^{r_i}$. It also computes $M = \prod_{i=1}^t M_i = g_1^{m_1 + \dots + m_t} h_1^{r_1 + \dots + r_t}$. Finally the principal sends σ, M, M_i for $1 \leq i \leq t$ to verifier.
- (2) *Zero-knowledge proof of aggregate commitment.* The principal and the verifier SP carry out the following ZKP protocol:

$$PK \left\{ (\alpha, \beta) : M = g_1^\alpha h_1^\beta, \alpha, \beta \in \mathbb{Z}_q \right\}$$

- (3) *Verification of aggregate signature.* After the verifier accepts the zero-knowledge proof of the commitments, it checks if the following verifications succeed:

$$M = \prod_{i=1}^t M_i \quad \text{and} \quad e(\sigma, g_2) = e(M, v)$$

Integrating the zero-knowledge proof into the verification



To prove the knowledge of secret commitments.

- (1) *Principal's aggregation.* Upon SP's requirement to prove $\sigma_1, \sigma_2, \dots, \sigma_t$, the principal aggregates the signatures into $\sigma = \prod_{i=1}^t \sigma_i$ where σ_i is the signature of committed value $M_i = g_1^{m_i} h_1^{r_i}$. The principal also computes $m = m_1 + \dots + m_t \pmod{q}$, $r = r_1 + \dots + r_t \pmod{q}$.
- (2) *Zero-knowledge proof of aggregate commitment.* The principal sends σ to SP, and carries out the following ZKP protocol with SP:

$$PK \{ (\alpha, \beta) : e(\sigma, g_2) = e(g_1, v)^\alpha e(h_1, v)^\beta, 0 < \alpha, \beta < q \}$$

Zero-knowledge proof the aggregated signature



To prove the possession of signature.

- (1) *Principal's aggregation.* Upon SP's requirement to prove a signature σ , principal chooses $r \in \mathbb{Z}_q$ at random, and sends the messages $\delta := \sigma^r$ to SP.
- (2) *Zero-knowledge proof of aggregate commitment.* The principal carries out the following zero-knowledge proof protocol with the verifier SP:

$$PK \{ (\alpha, \beta) : e(\delta, g_2) = e(g_1, v)^\alpha e(h_1, v)^\beta, 0 < \alpha, \beta < q \}$$



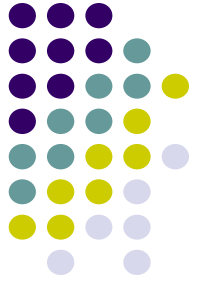
Efficiency Analysis

- Our signatures on commitments are short and the storage complexity is smaller than the ones computed with existing techniques [Camenisch et. Al.'04]
- Our approach is more flexible in that whenever n messages are committed for a user, the user is able to prove $2^n - 1$ many combinations of them which does not appear possible in the existing schemes

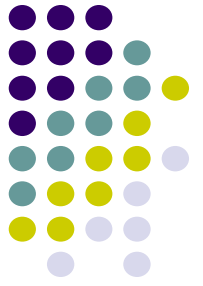
		Protocol 2	Protocol 3a	Protocol 3b	Protocol 4a	Protocol 4b
Our Protocols	provers	$2 + 2$	$3 + 2$	$3 + 2$	2	2
	verifiers	3	$2t + 3$	$2t + 3$	3	3
Without Aggregation	provers	$2t$	$4t$	$4t$	$2t$	×
	verifiers	$3t$	$5t$	$5t$	$3t$	×

Comparison of the number of exponentiations for proving t factors

Multi-factor Authentication using Aggregated Proof of knowledge

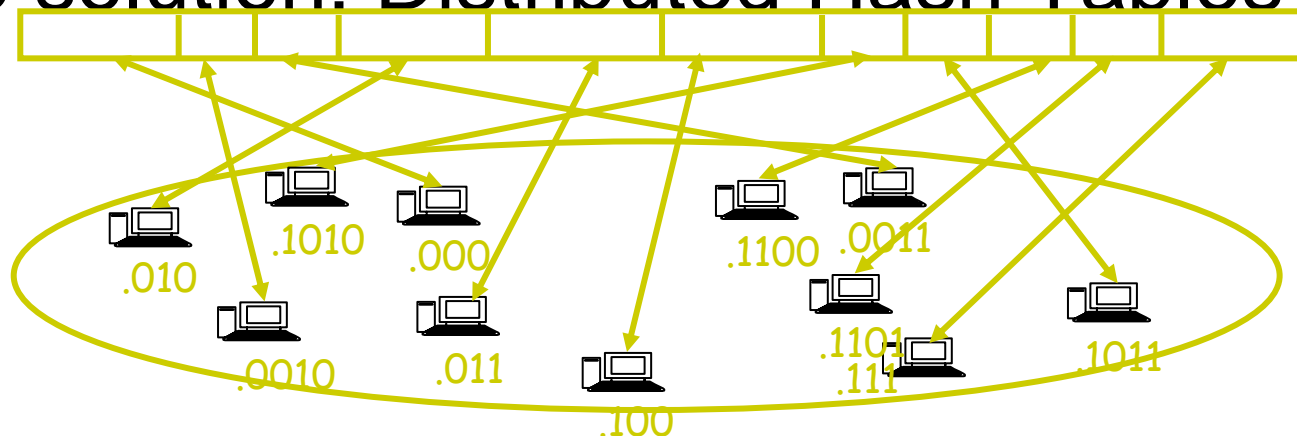


- Key Contributions:
 - New cryptographic primitive which provides methodologies for privacy preserving multi-factor authentication.
 - Computational efficiency - Reduces the proofs of several factors, that would require several Zero knowledge proofs of knowledge (ZKPK), to one that uses only one ZKPK.
 - Storage efficiency- Provides a flexible solution with minimal storage requirements.

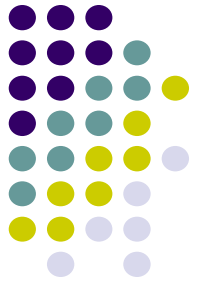


How to detect duplicates in a Federation?

- Put the strong identifiers in a hash table and look for collisions
- Problem: How can thousands of hosts cooperatively maintain a large hash table in a completely decentralized fashion?
- One solution: Distributed Hash Tables

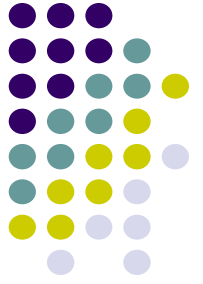


What are the main advantages of our solution?

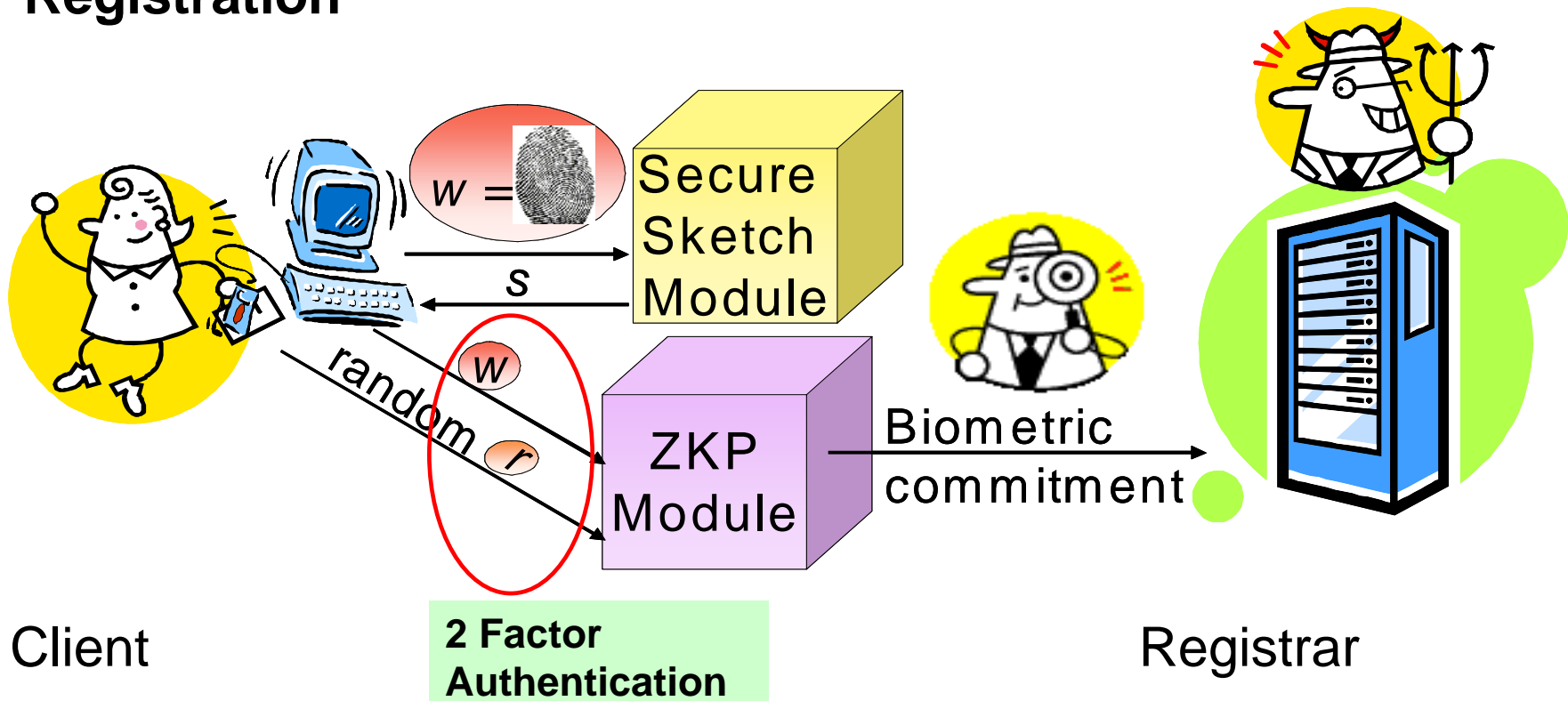


- The actual values of the registered attributes used as proofs for multi-factor authentication and privacy is secured using ZKP.
- Assurance of valid information in a federation.
- We allow a flexible approach to authentication and a novel lazy validation approach to information in the federation.

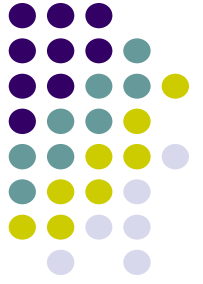
Combination with Biometric Authentication



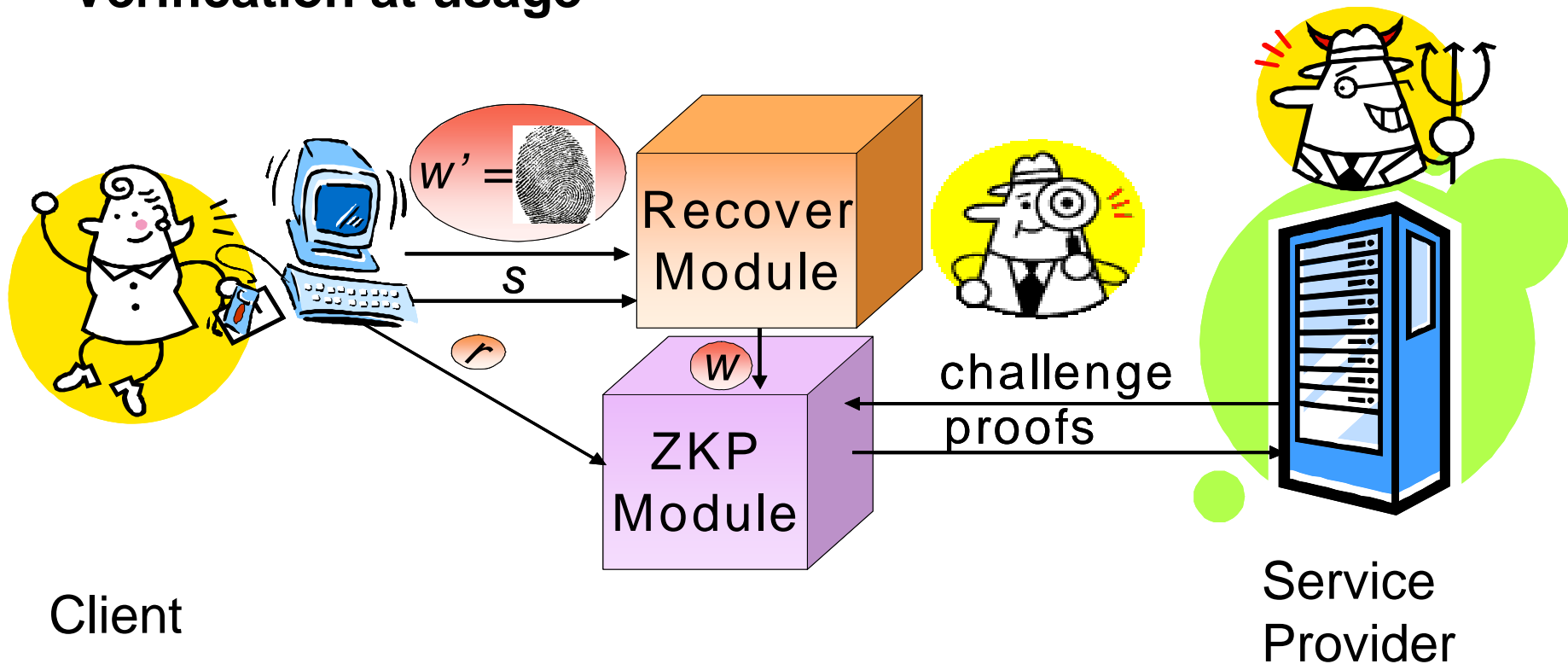
Registration



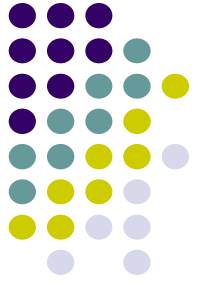
Combination with Biometric Authentication (cont.)



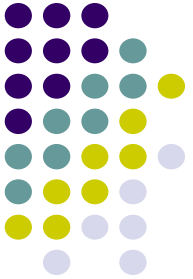
Verification at usage



Policies for Identity Management in Federations

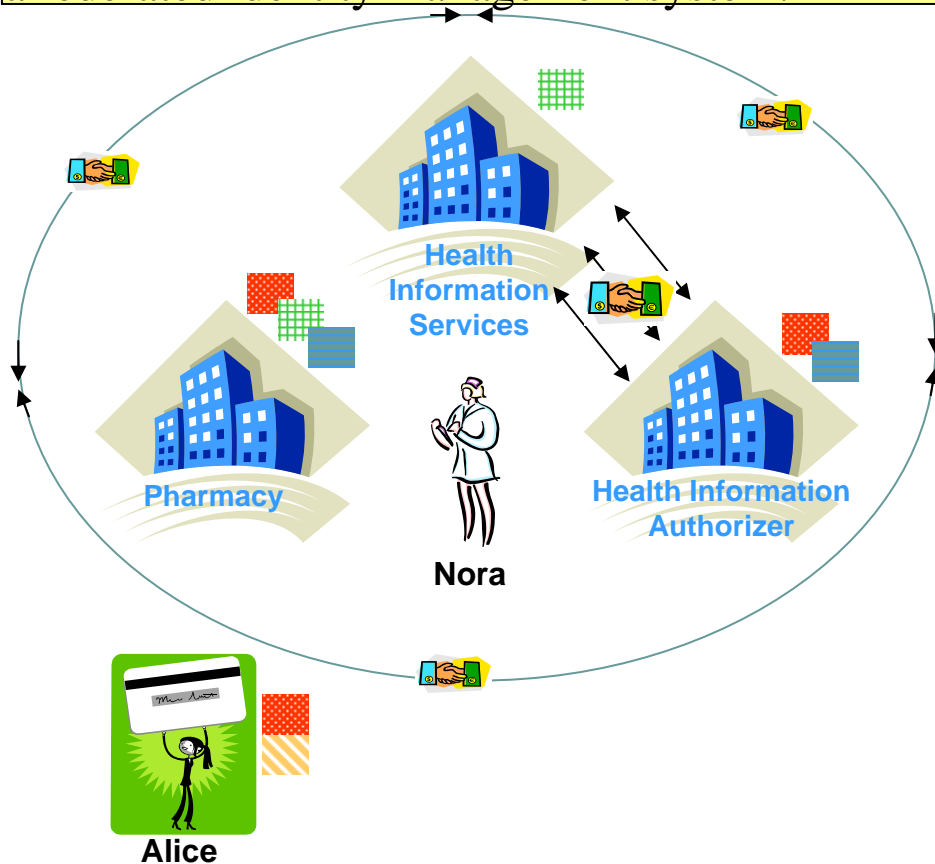







- We have developed a **comprehensive set of assertions** which is specifically relevant in the context of federations.
- Our assertions provide an **intuitive approach to model federation activities and make access control decisions based on a large variety of information**, including past access history.
- We analyze the history of the behavior of entities and events with the help of an assertion audit log and query processing, and also provide a simple approach to specify policies.



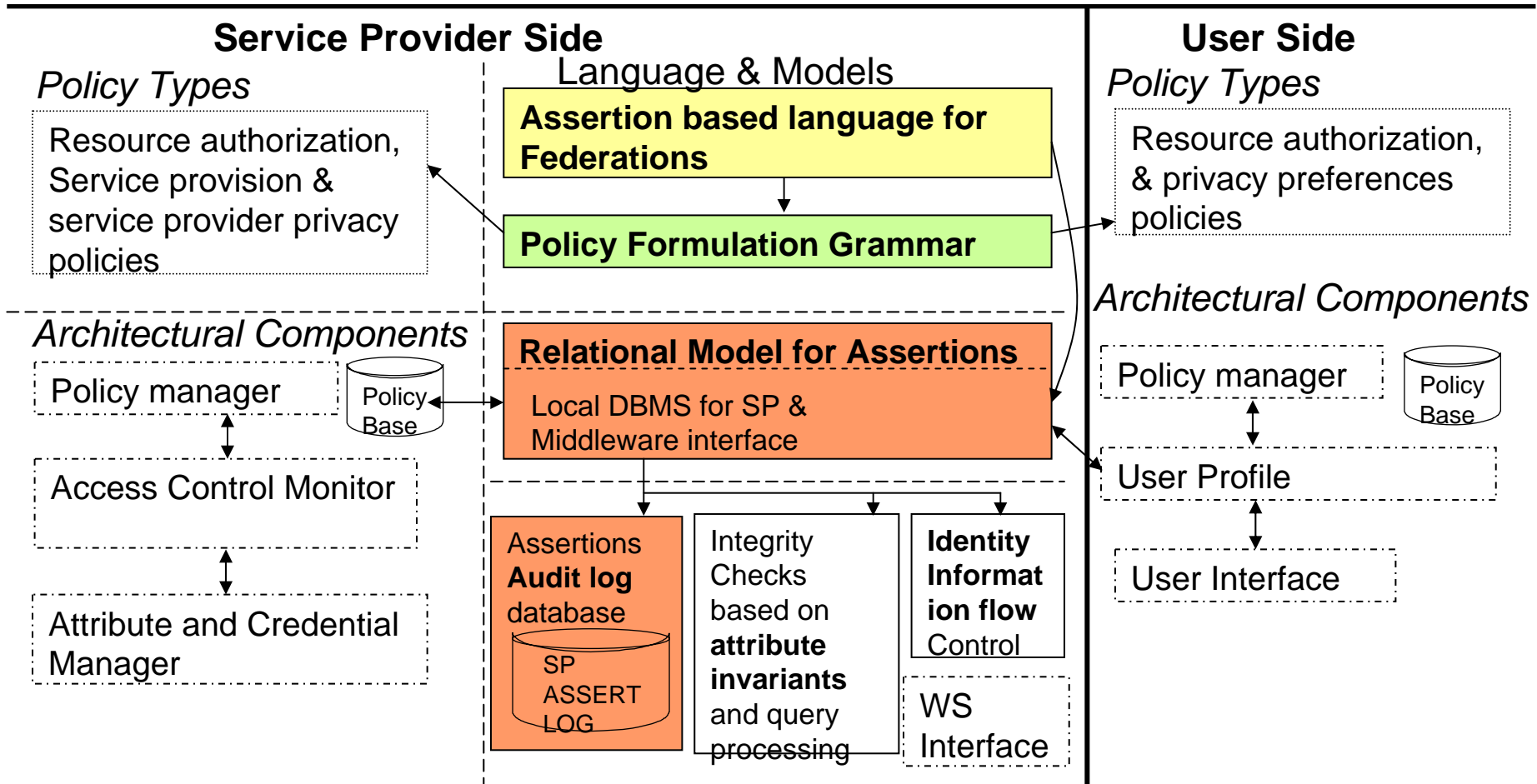
Policy for Managing Identities

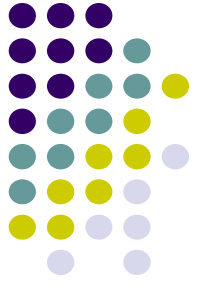
Managing identities have a lot of aspects. Therefore following is a taxonomy of policies in a federated identity management system.



	Authorization Policies
	Service Provision Policies
	Privacy Policies
	User Resources Preferences Policies
	Federation Agreement Policies

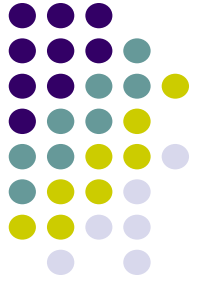
Assertion Based Policy Language for Federations





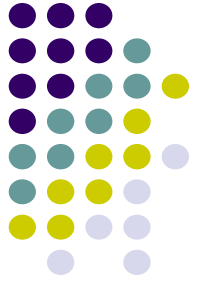
Assertions

- All actions taken by SP's and users for authorization can be described through assertions.
- *Each assertion is defined in terms of:*
 - The main interacting entities
 - A time-stamp
 - Other related information.
- The assertions capture the dynamic events occurring in the federation in a step by step, constructive approach.



Operational approach

- We propose to use a log of the actions executed by the entities in the federation;
- The log is a relational table, ASSERT_LOG defined according to the notion of relation of the relational data model.
- Checks for the log consistency are encoded using SQL-like queries.
- The log can be used to reason about the **flow of identity information** of the users.



Conclusion

- **Identity Management** and **Theft Protection** are areas of growing concern and active work.
- Identity Management system has potential to provide a secure and collaborative environment.
- We provide a solution to the problem of Identity Theft with the help of privacy preserving multi-factor authentication.

Thank You!

- Questions?
- Elisa Bertino
- bertino@cerias.purdue.edu

