

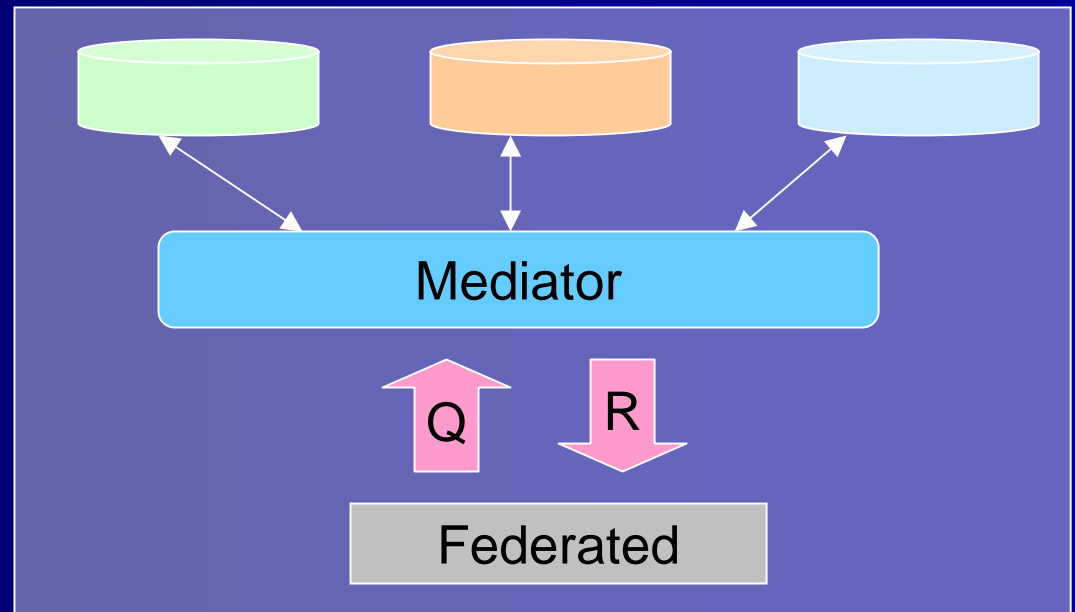
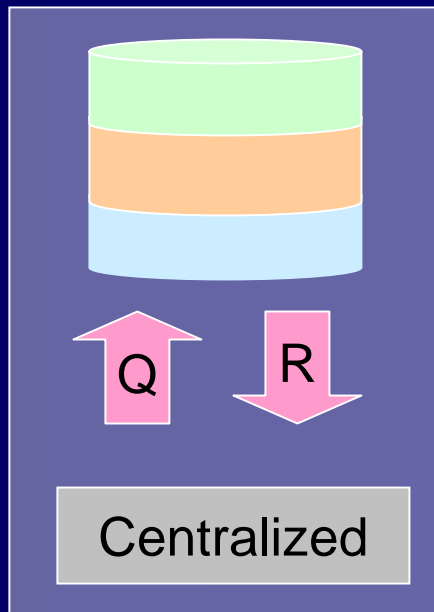
A Reusable Platform for Building Sovereign Information Sharing Applications

R. Agrawal, D. Asonov, P. Baliga, L. Liang, B. Porst R. Srikant
IBM Almaden Research Center

Outline

- Background
- Implementation Architecture
- Resource discovery, Schema mapping, and Authentication
- Performance
- Conclusion

Information Integration Today



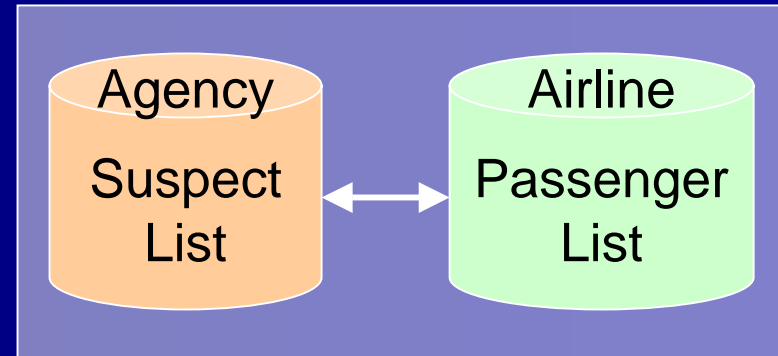
Assumption: Information in each database can be freely shared.

Need for a new style of information sharing

- Compute queries across databases so that no more information than necessary is revealed (without using a trusted third party).
- Need is driven by several trends:
 - End-to-end integration of information systems across companies (virtual organizations)
 - Simultaneously compete and cooperate.
 - Security: need-to-know information sharing

Security Application

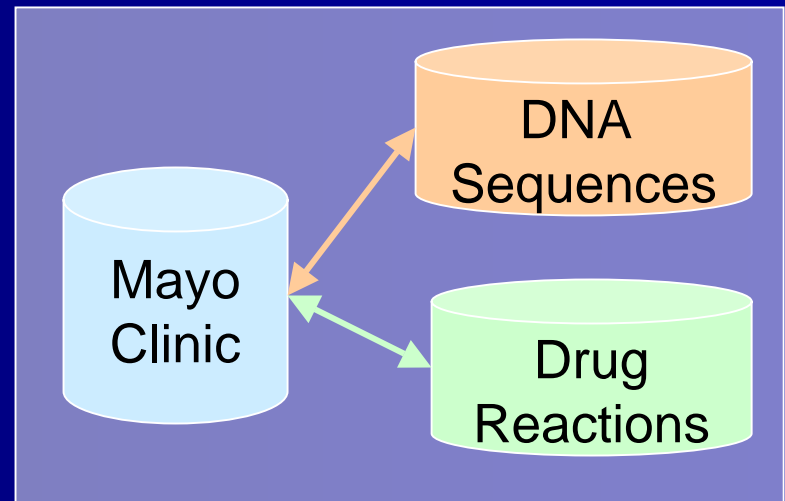
- Security Agency finds those passengers who are in its list of suspects, but not the names of other passengers.
- Airline does not find anything.



<http://www.informationweek.com/story/showArticle.jhtml?articleID=184010%79>

Medical Research

- Validate hypothesis between adverse reaction to a drug and a specific DNA sequence.
- Researchers should not learn anything beyond 4 counts:



	Adverse Reaction	No Adv. Reaction
Sequence Present	?	?
Sequence Absent	?	?

Minimal Necessary Sharing

R	
a	
u	
v	
x	

S	
b	
u	
v	
y	

$R \bowtie S$

- R must not know that S has b & y
- S must not know that R has a & x

$R \bowtie S$

u
v

Count ($R \bowtie S$)

- R & S do not learn anything except that the result is 2.

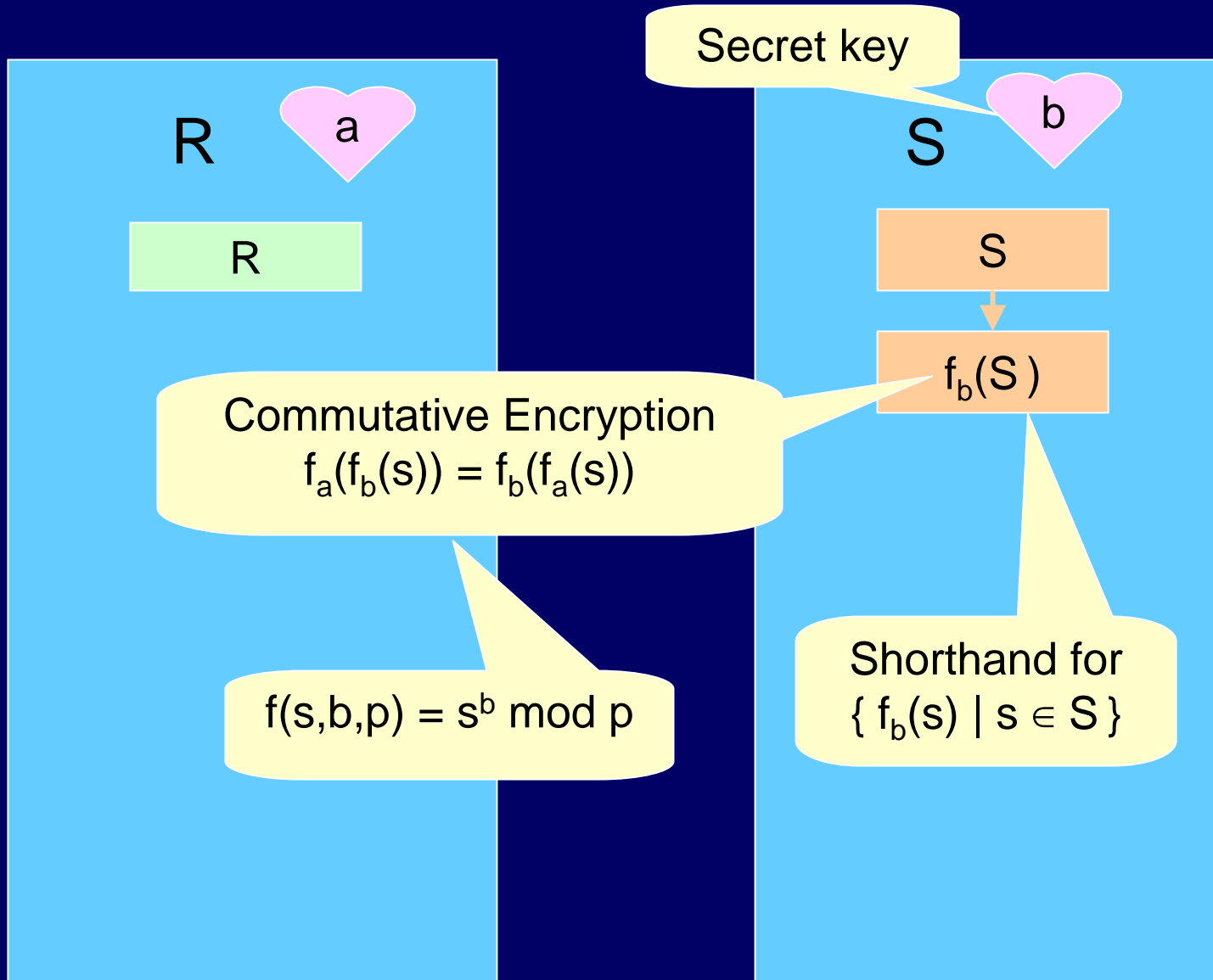
Problem Statement: Minimal Sharing

- Given:
 - Two parties (honest-but-curious): R (receiver) and S (sender)
 - Query Q spanning the tables R and S
 - Additional (pre-specified) categories of information I
- Compute the answer to Q and return it to R without revealing any additional information to either party, except for the information contained in I
 - For intersection, intersection size & equijoin,
 $I = \{ |R|, |S| \}$
 - For equijoin size, I also includes the distribution of duplicates & some subset of information in $R \cap S$

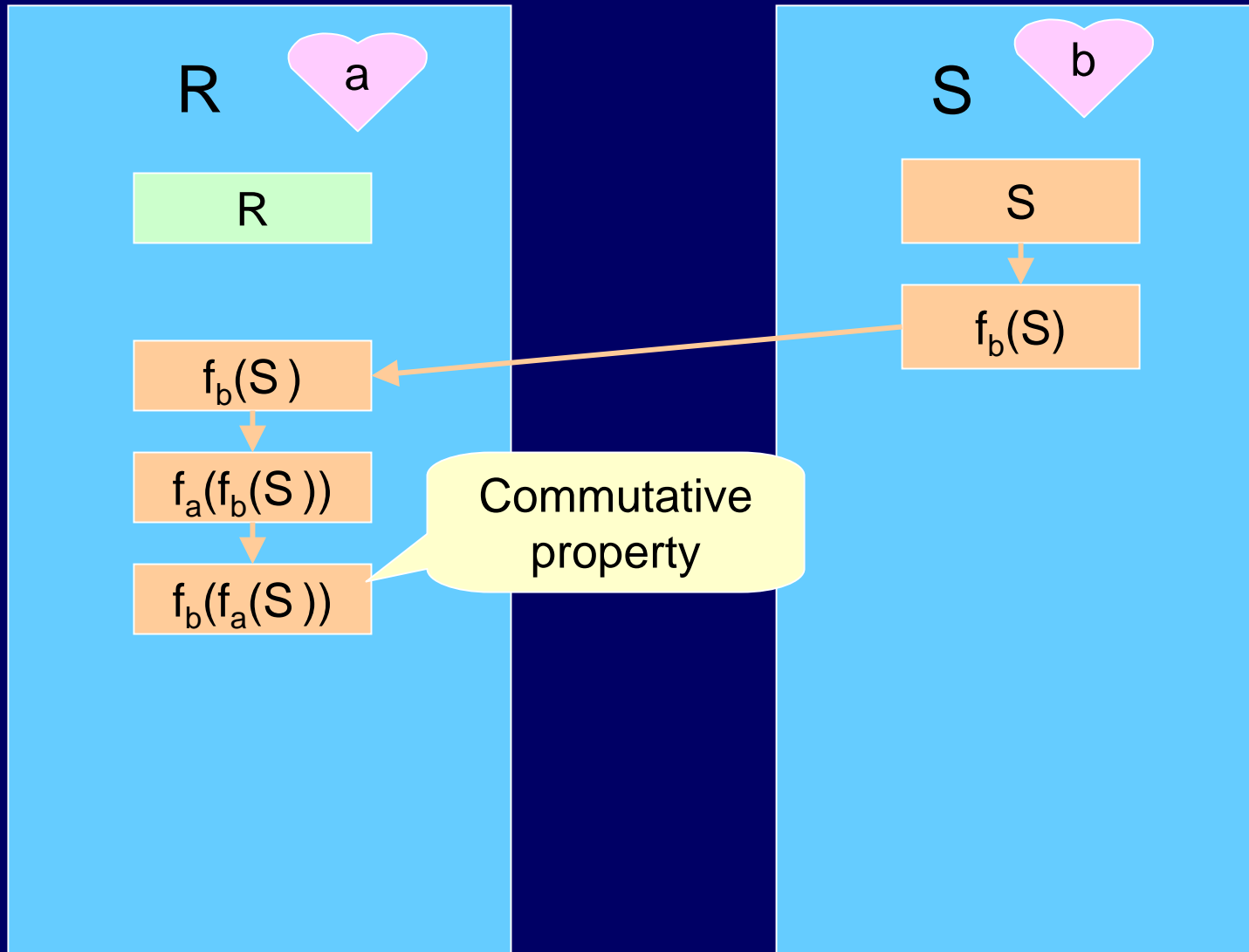
A Possible Approach

- Secure Multi-Party Computation
 - Given two parties with inputs x and y , compute $f(x,y)$ such that the parties learn only $f(x,y)$ and nothing else.
 - Can be solved by building a combinatorial circuit, and simulating that circuit [Yao86].
- Prohibitive cost for database-size problems.
 - Intersection of two relations of a million records each would require 144 days (Yao's protocol)

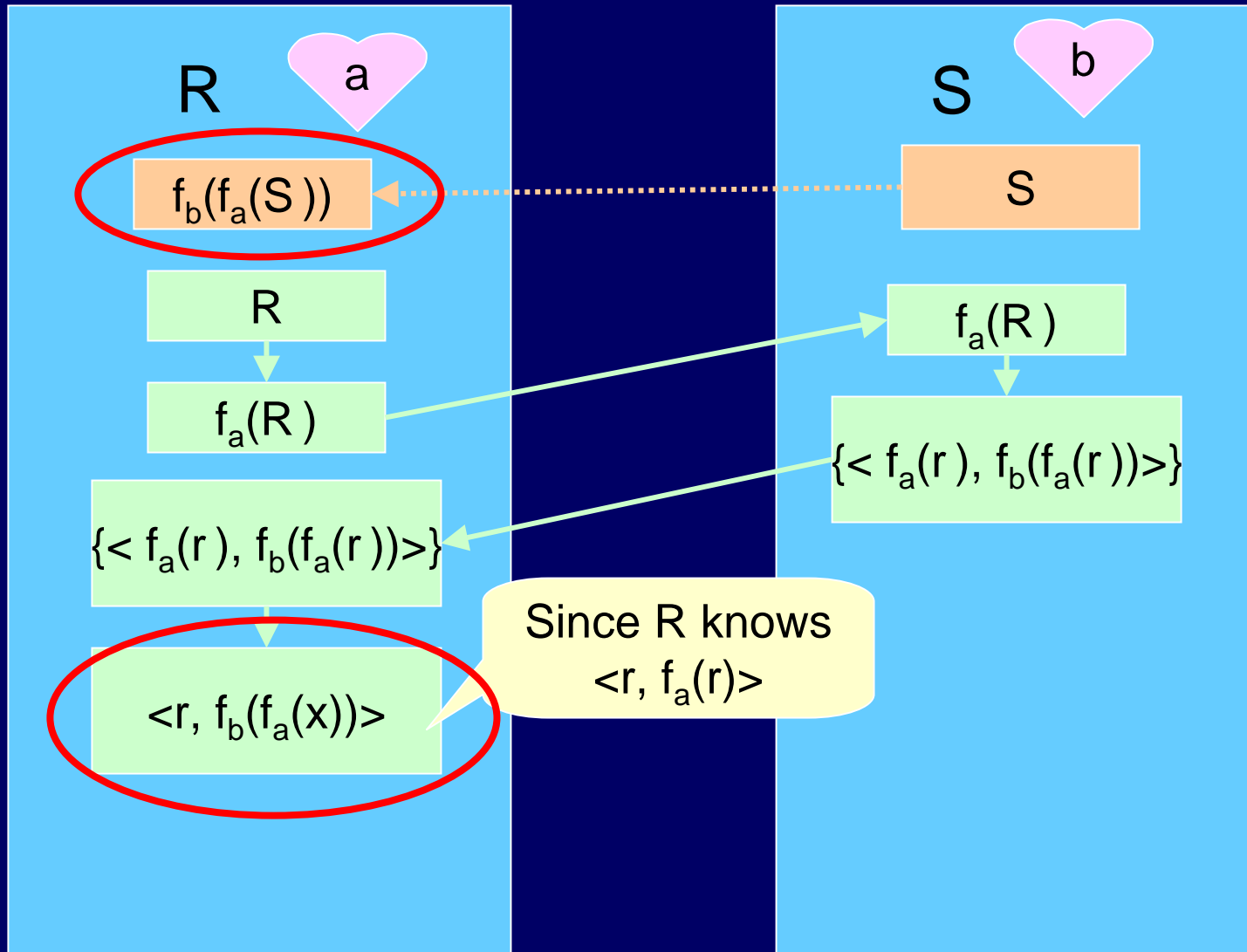
Intersection Protocol



Intersection Protocol



Intersection Protocol



Grid

Thin layer on top of the SIS client: invokes the required SIS operations, provides an interface to a SIS user.

Application Developer

Application

User

Mapping information and data provider access information.

Client Metadata

SIS Client

Constructs web service query requests against multiple data providers, and collects responses.

Provides the necessary functionality on the data provider side to enable sovereign sharing.

Includes view information to retrieve data from the data provider database, database access information, and context information.

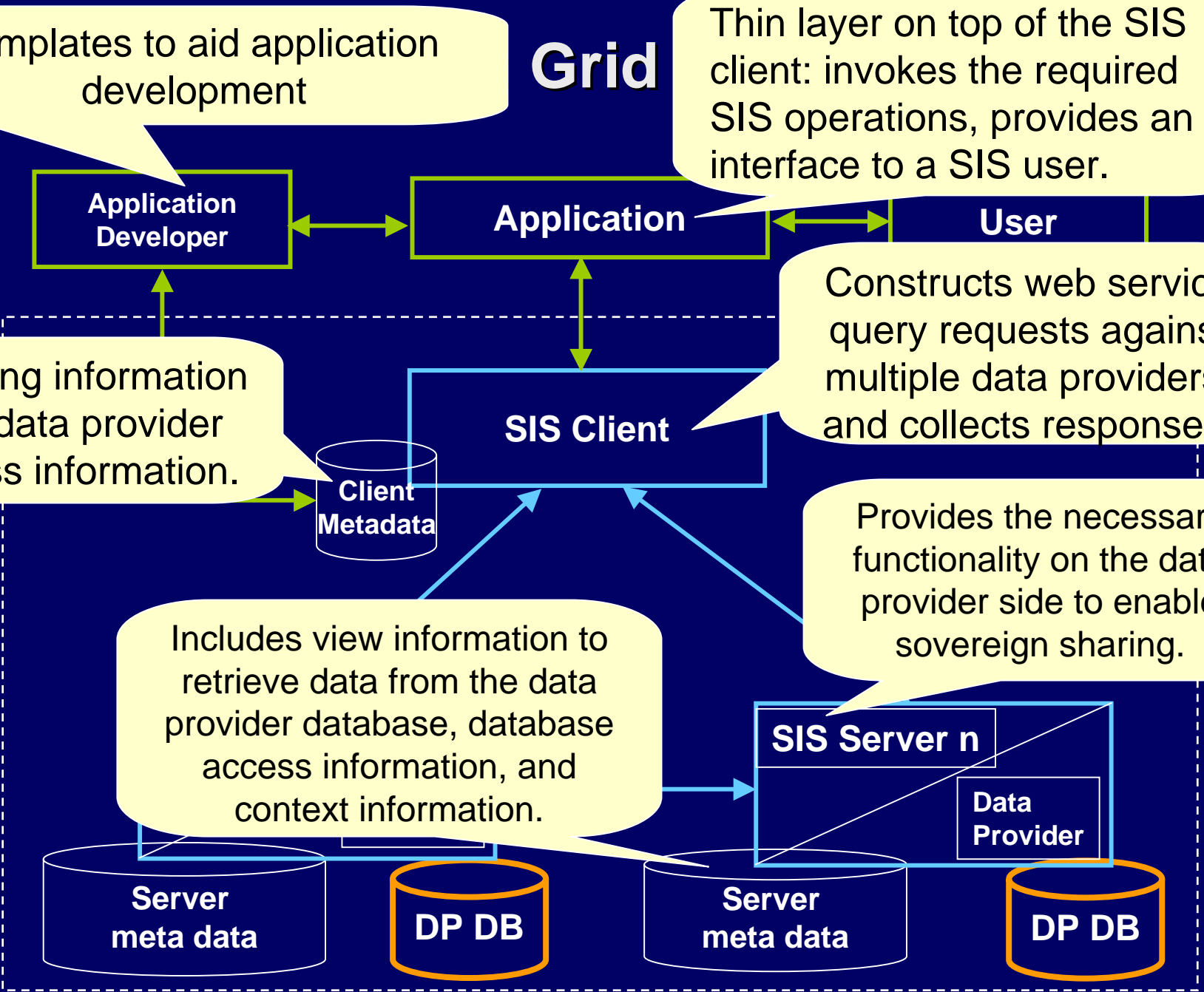
SIS Server n
Data Provider

Server meta data

DP DB

Server meta data

DP DB



Implementation Environment

- Data resides in DB2 v.8.1. database systems, installed on 2.4GHz/ 512MB RAM Intel workstations, connected by a 100Mbit LAN network.
- Web services run on top of the IBM WebSphere Application Server v.5.0 and use Apache AXIS v.1.1. SOAP library for messaging.
- IBM private UDDI registry installed on one of the machines.

Issues

- How does the application developer find the necessary data sources and their schemas?
 - need a *resource discovery* mechanism
- How does the application developer link the data between different providers?
 - need a *schema mapping* mechanism
- How to ensure that only eligible users can carry out the computation?
 - need an *authentication* mechanism

Resource Discovery

- Problem: Finding the necessary data sources and their schemas
- Solution: Employ a UDDI registry to store and search
 - data providers and operations they support
 - available schemas for each data provider
- TSA and AL publish their schemas in the Business Services elements of the private UDDI.

```
<Data Provider>  
  <Table Name="Passenger List"  
    Schema="AL">  
    <Column>  
      <Name>Passenger Name<Name>  
      ...
```

```
<Data Provider>  
  <Table Name="Suspect List"  
    Schema="TSA">  
    <Column>  
      <Name>Suspect Name<Name>  
      ...
```

Schema Mapping

1. Global:

- Data providers use a standard domain-specific vocabulary (e.g. Rosetta Net) and schema.
- Data providers map local schema to global schema.

2. Application Specific:

- Every data provider maps local schema to the application schema, separately for each application.
- Every data provider updates its mapping as the application evolves.

3. Sovereign:

- Data providers publish schemas in their own vocabularies.
- Developers link the schemas.

Schema Mapping

1. Global:

- Data
- (e

2. App

- E
- S
- Ev
- evolves.

- Least burden on data providers
- Maximal autonomy for data providers and developers

3. Sovereign:

- Data providers publish schemas in their own vocabularies.
- Developers link the schemas.

Schema Mapping: TSA-AL scenario

The application developer determines mapping (possibly negotiating using the information in the Business Entity element of the UDDI registry)

Airline schema

TSA schema

<Data Provider>

<Table Name="Passenger List"

Schema="AL">

<Column>

<Name>Passenger Name<Name>

...



<Data Provider>

<Table Name="SuspectList"

Schema="TSA">

<Column>

<Name>Suspect Name<Name>

...



Authentication Across Multiple Domains



Application receives an authentication token from AA



User: Bob
Certified by AA

User: Bob
Certified by AA



Username: Bob
Password: ****

4

2

3

1

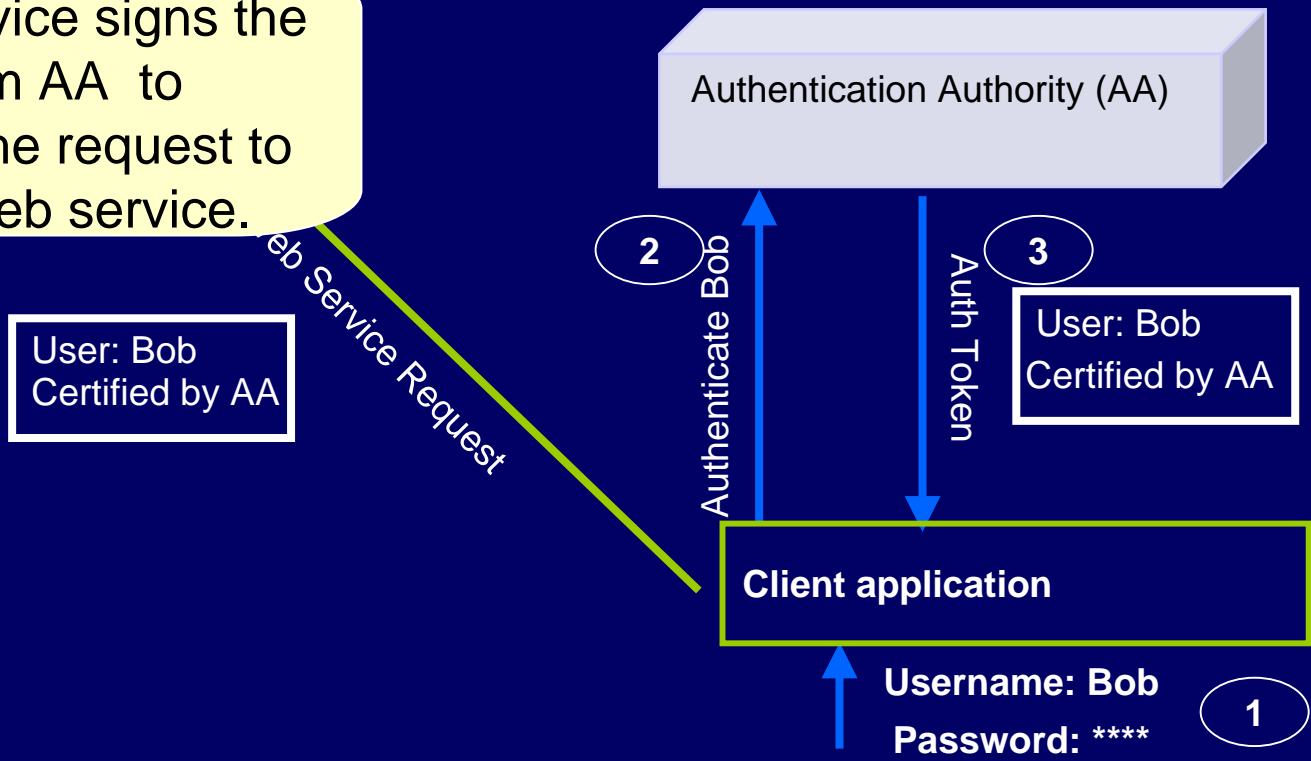
Web Service Request

The token is used to authenticate the client application to the client web service.

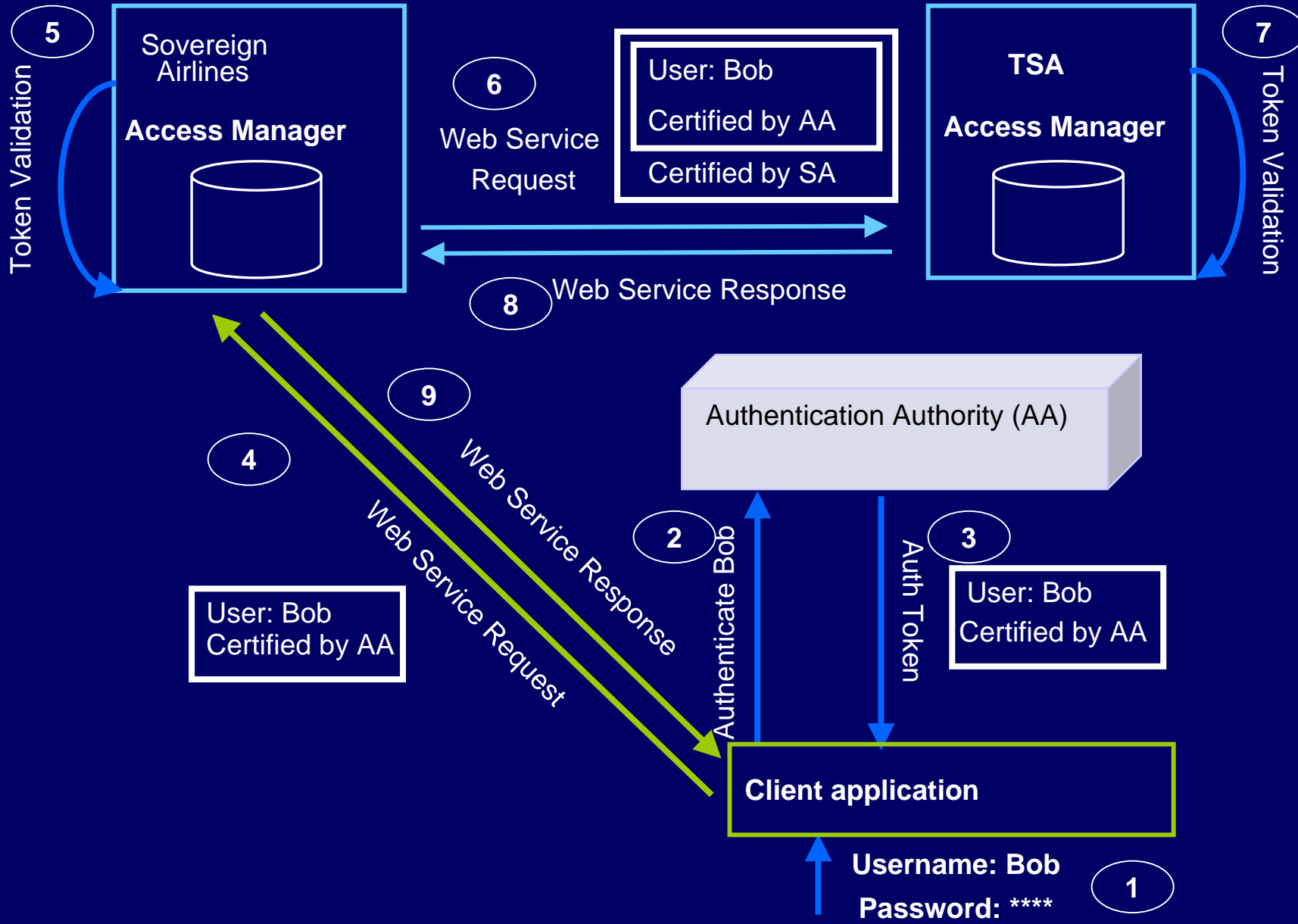
Authentication Across Multiple Domains



Client web service signs the token from AA to authenticate the request to the server web service.



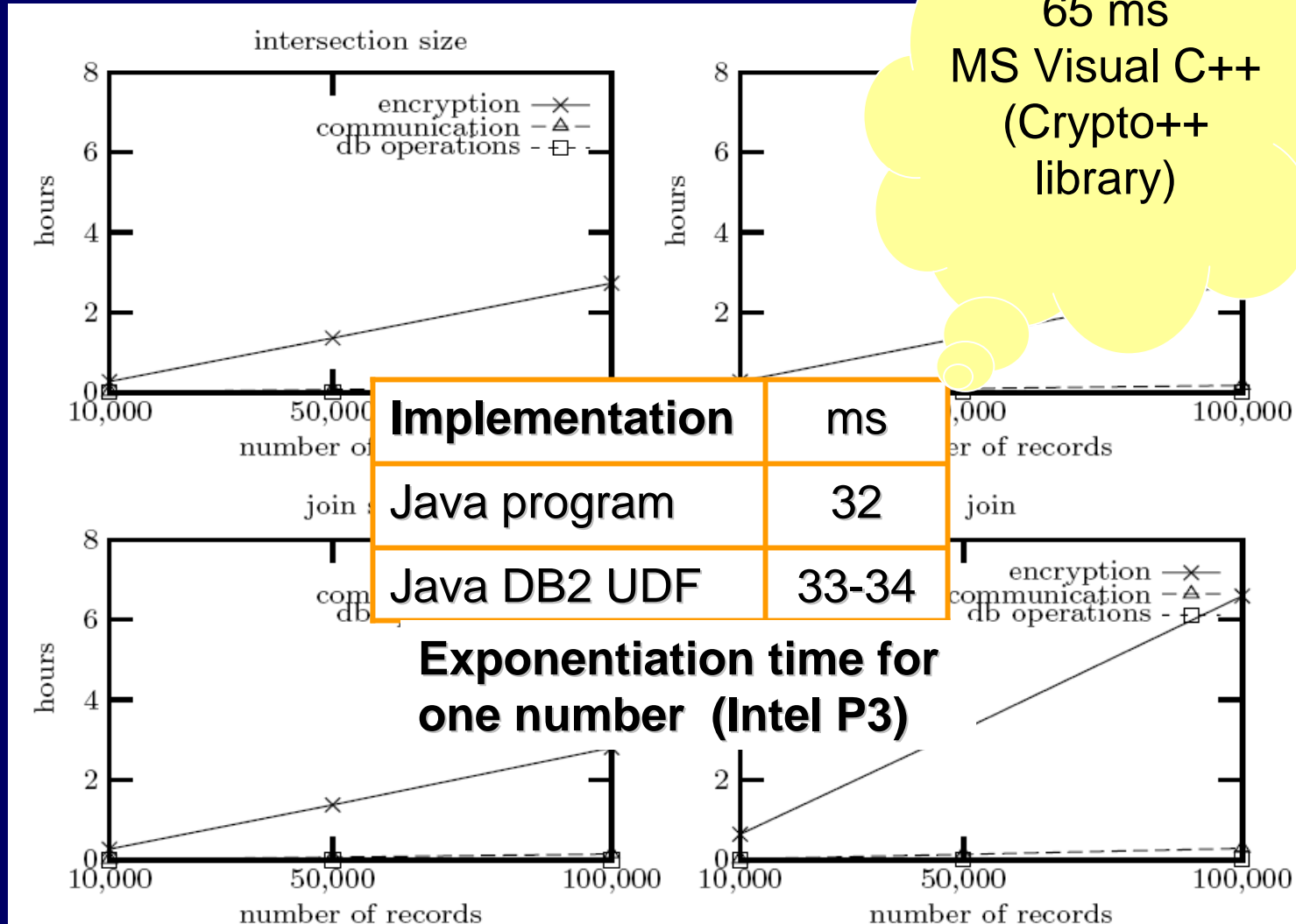
Authentication Across Multiple Domains



Security Application: Execution

- TSA encrypts *Suspect Name* column of *Suspect List* table.
- TSA sends an intersection web-service request to AL, with the encrypted *Suspect List* as a SOAP attachment.
- AL encrypts *Passenger Name* column of *Passenger List* table and double-encrypts the *encrypted Suspect List* from TSA.
- AL sends a web-service response to TSA with both encrypted tables as attachments.
- TSA double-encrypts Passenger List from AL.
- Finally, TSA uses both double-encrypted tables to perform the intersection and returns the results to the application.

Performance



Making Encryption Faster: Software Approaches

- The main component of encryption is exponentiation: $enc(x, k, p) = x^k \text{ mod } p$
- Tried custom implementations of exponentiation that used preprocessing based on
 - fixed exponent (k)
 - fixed base (x)
- Fixed exponent implementation turned out to be slower than the Java native implementation
- Fixed based is beneficial if the same value is encrypted multiple times with different keys (not useful for intersection where each value is encrypted once)

Making Encryption Faster: Hardware Accelerator

- Use SSL card to speed-up exponentiation
- Multiple threads (100+) must post exponentiation request simultaneously to the card API to get the advertised speed-up
- AEP scheduler distributes exponentiation requests between multiple cards automatically; linear speed-up



Example:
AEP SSL CARD Runner 2000
≈ \$2k

Execution time: Encryption UDF

Encryption Engine	Number of rows in the table		
	1,000	5,000	10,000
CPU Intel III 2.0 Ghz	34s	175s	320s
AEP Runner 2000	3.5s	19s	37s

Application Performance

- Encryption speed is 20K encryptions per minute using one accelerator card (\$2K per card)
- TSA-Airline: 150,000 (daily) passengers and 1 million people in the watch list:
 - 120 minutes with one accelerator card
 - 12 minutes with ten accelerator cards
- Epidemiological research: 1 million patient records in the hospital and 10 million records in the Genbank:
 - 37 hours with one accelerator cards
 - 3.7 hours with ten accelerator cards

Related Work

- [Naor & Pinkas 99]: Two protocols for list intersection problem
 - Oblivious evaluation of n polynomials of degree n each.
 - Oblivious evaluation of n^2 linear polynomials.
- [Huberman et al 99]: find people with common preferences, without revealing the preferences.
 - Intersection protocols are similar
- [Clifton et al, 2003]: Secure set union and set intersection
 - Similar protocols

Summary and Challenges

- New applications require us to go beyond traditional centralized and federated information integration: sovereign information integration
- Demonstrated feasibility of realizing sovereign sharing
- Need models of minimal disclosure and corresponding protocols for
 - other database operations
 - combination of operations
- Need faster commutative encryption
- Need further study of tradeoff between efficiency and
 - additional information disclosed
 - approximation