Database & Storage Security

Professor Dr. Mohammad Abu Yousuf
yousuf@juniv.edu

#Lecture-3.2

Database issues in Trust Management & Trust Negotiation

Trust Management – Basic

•*Trust Management, introduced by Blaze et al. [BFL96], is a unified approach* to specifying and interpreting security policies, credentials, and relationships that allows direct authorization of security-critical actions.

•In particular, a trust management system combines the notion of **specifying security policy** with the mechanism for specifying **security credentials**.

•Credentials describe specific delegations of trust among public keys; unlike traditional certificates, which bind keys to names, trust-management credentials bind keys directly to authorizations to perform specific tasks.

Trust Management – Basic

- A traditional "**system-security approach**" to the processing of a signed request for action treats the task as a combination of *authentication and access control.*
- The receiving system first determines ***who signed the request*** *and* ***then queries*** **an internal database to decide** ***whether the signer should be granted access to*** **the resources needed to perform the requested action.**
- Question- "who signed this request?"
- The *trust management approach, initiated by Blaze et al. [BFL96], frames the question as* follows: **"Does the set *C of credentials prove that the request r complies with* the local *security policy P?"***

Trust Management – Physical World

- **Certified credentials** form the BASE of trust.
- Citizens identify themselves
  - at the voting booth with national **identity cards**,
  - motorists demonstrate their right to drive cars with **driver licenses**,
  - customers pay for their groceries with **credit cards**,
  - airline passengers board planes with their **passports** and **boarding passes**, and
  - sport enthusiasts make their way into the gym using their **membership cards.**
- Often such **credentials** are used in contexts beyond what was originally intended:
  - for example, **identity cards** are also used to prove eligibility for certain social benefits, or to demonstrate to be of legal age when entering a bar.

Trust Management – Physical World

- Each of these **credentials** contains attributes that describe
  - the **owner** of the credential (e.g., name and date of birth),
  - the **rights** granted to the owner (e.g., vehicle class, flight and seat number),
  - or the credential itself (e.g., expiration date).
- The information in the credentials is trusted because it is certified by an issuer (e.g., the government, a bank) who on its turn is trusted.
- Hence the **credential** works well in trust management.

Trust Management – Physical World

What these credentials (certificates) carry?

- **Identity** of the subject.
- Only **authorization** is required to have access
  - Train/Bus Ticket
  - University ID card to get access in the main gate of the campus.
  - Cinema ticket, cricket match ticket
- Both **Identity** & **Authorization**
  - Student entry in the exam hall.
  - To obtain Boarding pass in the plane.
- Identity of the issuer.

Trust Management – Physical World

How Trust management occurs in this world?

- Half ticket fare for students of train and bus
  - National ID + Student ID
- Free ticket for Bangladeshi students to access to zoo
  - National ID + Student ID
- Free access to all nationally arranged games and sports for Bangladeshi athletes
  - National ID + Athletes ID from games and sports authority.
- Free access to all public libraries for the Bangladesh students.
  - National ID + Student ID

Trust Management – Physical World

Analysis of the traditional TM activities:

- **Resource Access Policy:**
  - It governs the access rights to the resources.
  - It lies in the hands of resource owner.
  - They decide which certificate (Passport, National ID, Student ID etc.) they will use for which resources.

Trust Management – Physical World

Analysis of the traditional TM activities:

- Security credential:
  - It certifies (binds) the identity(name, address, Serial no etc.) of the resource requester.
  - The certificates need to be verified by the resource owner.
    - Only physical verification (signature, seal, logo etc.) on the spot.
    - Chance of fabricated certificate.

Trust Management – Physical World

Analysis of the traditional TM activities:

- Certificate Authority:
  - Some third parties (Passport Office, Election Commission, University authority etc.) produces and distributes the security credentials.
  - They certifies the identity only of the user of the certificates.
  - They do not bear the responsibility of a certificate holder's fraudulent transactions.

What is Trust Management System?

- A mechanism where authorization decision is taken on the fly in the Internet with no pre-established relationship.

- Uses cryptographic credentials (PGP, X.509) to convey information relevant to authorization.

- Authorization decision: whether the information contained in a given set of credentials are sufficient to access the requested resource as per the current policy governing that resource.

What is Trust management?

- In TM system, security policy is made by local administrators to specify access control rules on local resources.

- Credentials in TM system usually do not typically bind public keys to identities.

- It binds other information on which authorization decisions are based.

What is Trust management?

- Credentials may also grant the right to further delegate the capability.

- Chains of such credentials can be used to document a sequence of delegations of privileges from the resource owner to the requester.

Steps of credential based TM Application

1. Obtain certificates, verify signatures on certificates and on application request, determine public key of original signer(s).
2. Verify that certificates are unrevoked.
3. Attempt to find "trust path" from trusted certifier to certificate of public key in question.
4. Extract names from certificates.
5. Lookup names in database that maps names to the actions that they are trusted to perform.
6. Determine whether requested action is legal, based on the names extracted from certificates and whether the certification authorities are permitted to authorized such actions according to local policy.
7. Proceed if everything appears valid.

Steps of TM System based Application

1. Obtain certificates, verify signatures on certificates and on application request, determine public key of original signer(s).
2. Verify that certificates are unrevoked.
3. Submit request, certificates, and description of local policy to local "trust management engine".
4. Proceed if approved.

Three Components of Trust Management

- **Security policies** which are local trust assertions that the local system trust unconditionally.

- **Security Credentials** which are signed trust assertions made by other parties.

- **Trust Relationship** – issuance of certificates by any parties initiates the trust relationships.

Purpose of Trust Management

– A mechanism that allows one to decide which requesters are qualified to gain access to the resource.

– on the other hand, which server is trusted to provide the requested service, on the basis of certified statements provided by the interacting parties is needed.

Trust management in digital world

- Managed through Access Control
  - Authorization Policy
    - Various privileges/roles
    - Assigning userID with these privileges/roles.
  - Identification and authentication
    - Log in with Username & Password and its verification
  - Authorization
    - Matching userID with the authorization policy.

Trust management in digital world

- Within a single organization, pre-established trust relationships are used to assign authorizations.
- Among the organizations of the trading partners, pre-established trust relationship increases administrative burden of managing a large number of users of all organizations.
- It becomes very difficult to manage the trust between two users when there is no pre-established trust relationship.

Trust management in digital world

- The complexity culminates when there is a need for anonymity of users (electronic voting).

- A new technique is required that enables on-line parties to establish trust on the fly.

Trust management in digital world

- **Trust Management with TM System**
  - Most are based on public key "certificates" in which a trusted 3rd signs a specially formed message certifying the **identity** associated with a **public key**.
  - How the certified identity is acted upon is left to the application.
  - Two best known certificate systems are
    - PGP
    - X.509

Trust Management using PGP certificates

How does PGP works?

- Pretty Good Privacy uses a variation of the public key system. In this system, each user has an encryption key that is publicly known and a private key that is known only to that user.

- You encrypt a message you send to someone else using their public key. When they receive it, they decrypt it using their private key.

- .

Trust Management using PGP certificates

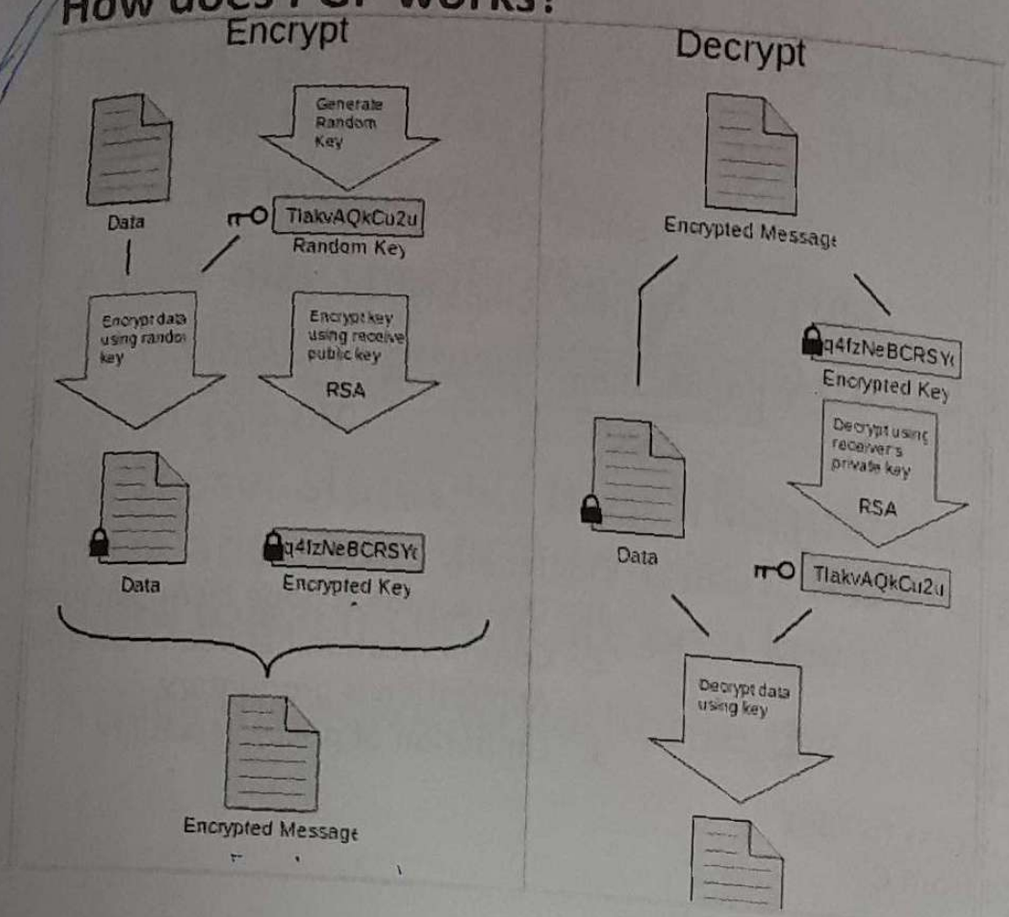
How does PGP works?

- Since encrypting an entire message can be time-consuming, PGP uses a faster encryption algorithm to encrypt the message and then uses the public key to encrypt the shorter key that was used to encrypt the entire message.

- Both the encrypted message and the short key are sent to the receiver who first uses the receiver's private key to decrypt the short key and then uses that key to decrypt the message.

Trust Management using PGP certificates

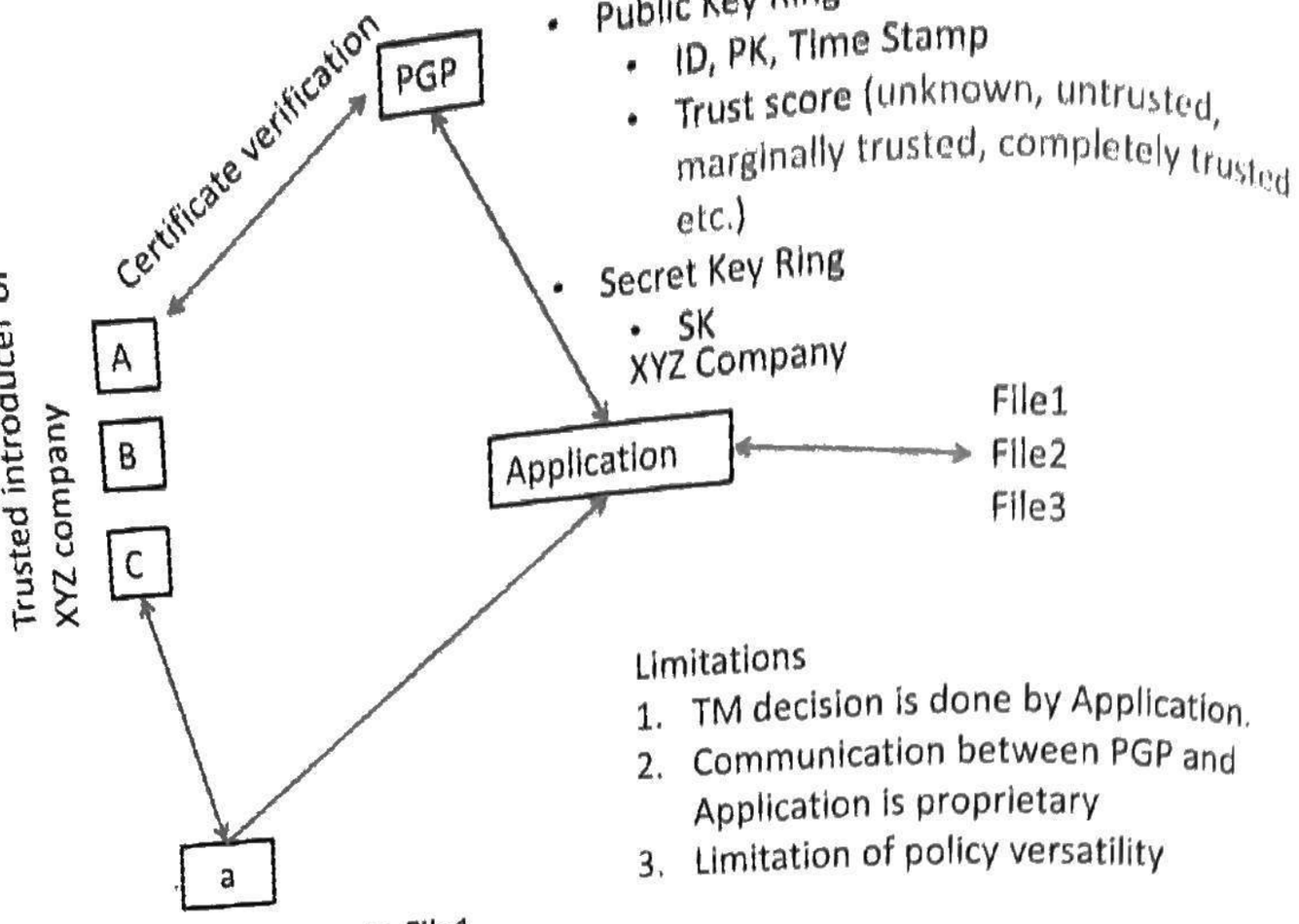
How does PGP works?



Trust Management using PGP certificates

- The combination of the two encryption methods combines the convenience of public key encryption with the speed of conventional encryption.
- Conventional encryption is about 1, 000 times faster than public key encryption. Public key encryption in turn provides a solution to key distribution and data transmission issues.
- Used together, **performance and key distribution are improved without any sacrifice in security.**

Trust Management using PGP certificates



Trust Management using PGP certificates

- Trust Management using PGP certificates
  - A user generates a key pair (PublicKey, SecretKey) that is associated with his unique ID (Name, Email Address)
  - Keys are stored key records with timestamp:
    - **Public Key Ring:** All public keys
    - **Secret Key Ring:** Own secret key.
  - User A has a good copy of user B's public key record.
  - User A can sign this copy (Key Certificate) and pass it to C. Thus A acts as an introducer.

Trust Management using PGP certificates

- Each user must tell the PGP system which individuals he or she trusts as introducers and must certify the introducers' public key records with his own secret key.
- A user may specify degree of trust:
  - Unknown, untrusted, marginally trusted, completely trusted etc.
- Each user stores his trust information on his key rings and tunes PGP so that it assigns a validity score to each certificate on a key ring.
- It uses the score to authorize the transaction.

Trust Management using PGP certificates

- **Security policy:** Verification of ID of sender.
- Key rings and degrees of trust allow each user to design his own security policy of very limited form.
- PGP is for secure email communication.
- No certificate authority.
- No centralized trust servers for certificate distribution.

PGP Certificate

- The user chooses how to use the certificate.
  - User C might be confident about A's trustworthiness and accept B's certificate, which A has signed.
  - A pessimistic user might only accept certificates certified by fully trusted entities, whereas an optimistic user might trust marginally trusted signers.

Trust Management using X.509 certificates

- In X.509, the trusted third party is a certificate authority (CA), which is usually a trustworthy entity for issuing certificates (Verisign, for example).
- Another CA might also certify a particular CA.
- When a user generates a **public/private** key pair, it registers its public key with a CA and has the CA certify it.

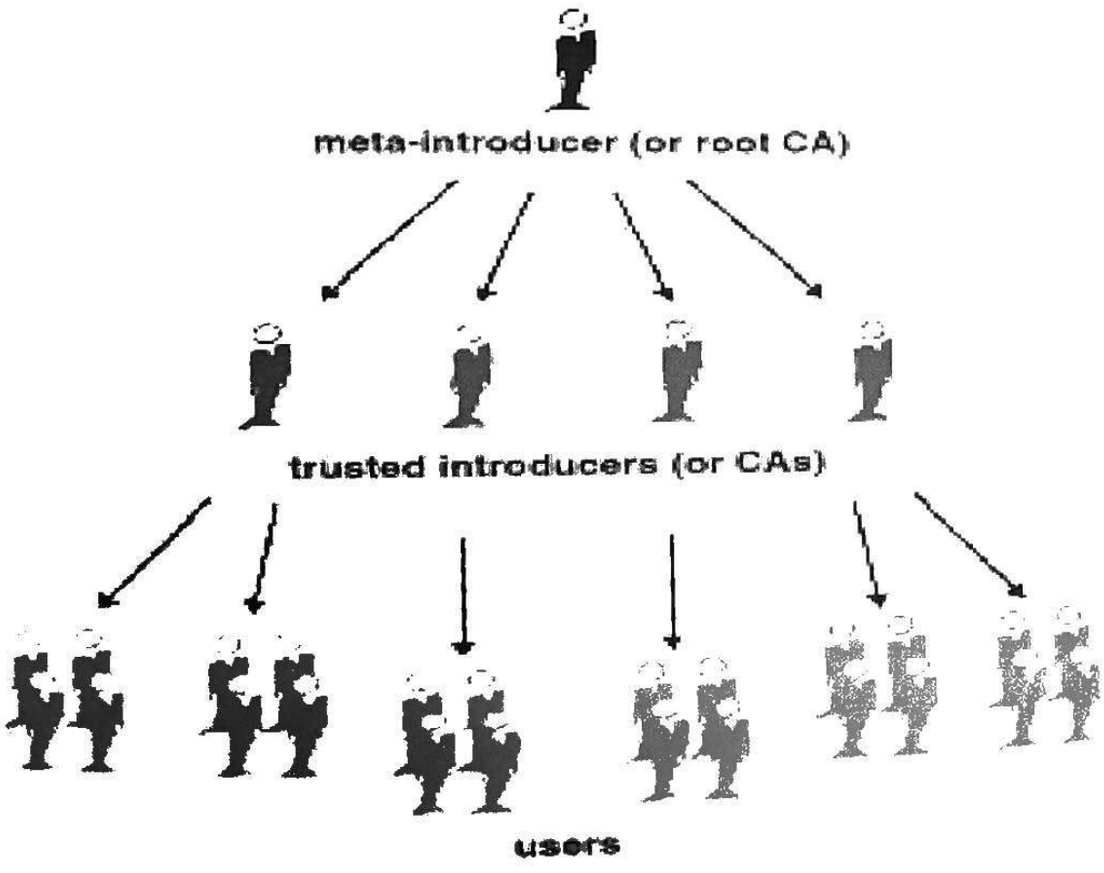
Trust Management using X.509 certificates

- If the same CA certifies two users and they want to communicate securely, they need only exchange their certificates.

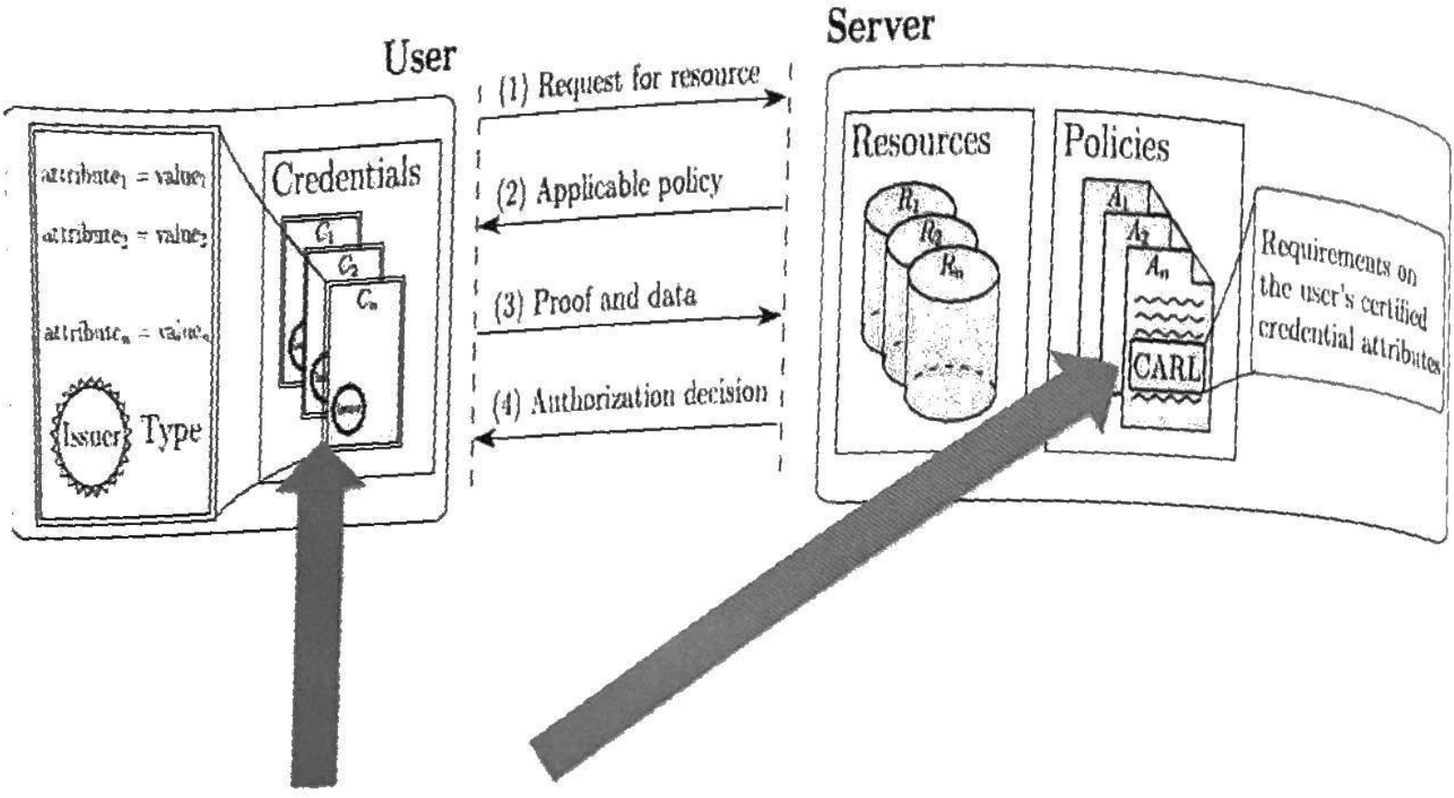
- If different CAs certify two users, they must resort to higher-level CAs, which certify their CAs until they reach a common CA.

- So, X.509 uses a hierarchical structure, which constructs a tree of trust.

X.509 Certificate



Credential Based Access Control



Raw material: Credentials & Policy

Problems of TM based on PGP & X.509

- Application specific.
- Identity based certification that is used in trust management.
- Security policy can not be made versatile.
- Trust management is done by the Application itself.
- No further modification.

- Seminal works on this issue:
  - Bina et al. proposed using characteristics other than identity, attested to by known authorities in digital certificates, as a basis for authorization on the Internet
  - Blaze et al. introduced a complementary approach to authorization based on delegation of privileges.
  - Rivest et al. introduced a scheme that provides a way to introduce names and bind them to public keys controlled by individuals and groups, which greatly facilitates identifying authorized principals electronically.

Challenges in TM Process

- Credentials are issued in a **decentralized** manner and somehow the relevant **credentials** need to be **collected** or otherwise made available to the **authorization** evaluation process.

- **Some** credentials carry **sensitive, confidential information** and may need to be subject to access **control themselves.**

Challenges in TM Process

- An access control policy may give clues about the nature of the resources it protects.

- If a patient's prescription can be viewed only by their pharmacist or by their parent, then one can guess that the prescription is for a child.

- To preserve the privacy of the resources that they protect, policies themselves may need protection.
- In other word, access to the contents of a policy may need to be governed by another access control policy.

Parties in Trust Management

- **Certificate Authority**
  - Cryptographic credentials (X.509, PGP)
  - Digital Certificate & Digital Signature
- **Resource requester**
  - Cryptographic credentials (X.509, PGP)
  - Digital Certificate & Digital Signature
- **Resource Provider**
  - Cryptographic credentials (X.509, PGP)
  - Digital Certificate & Digital Signature
  - Resource
  - Security Policy that governs the access to the resource

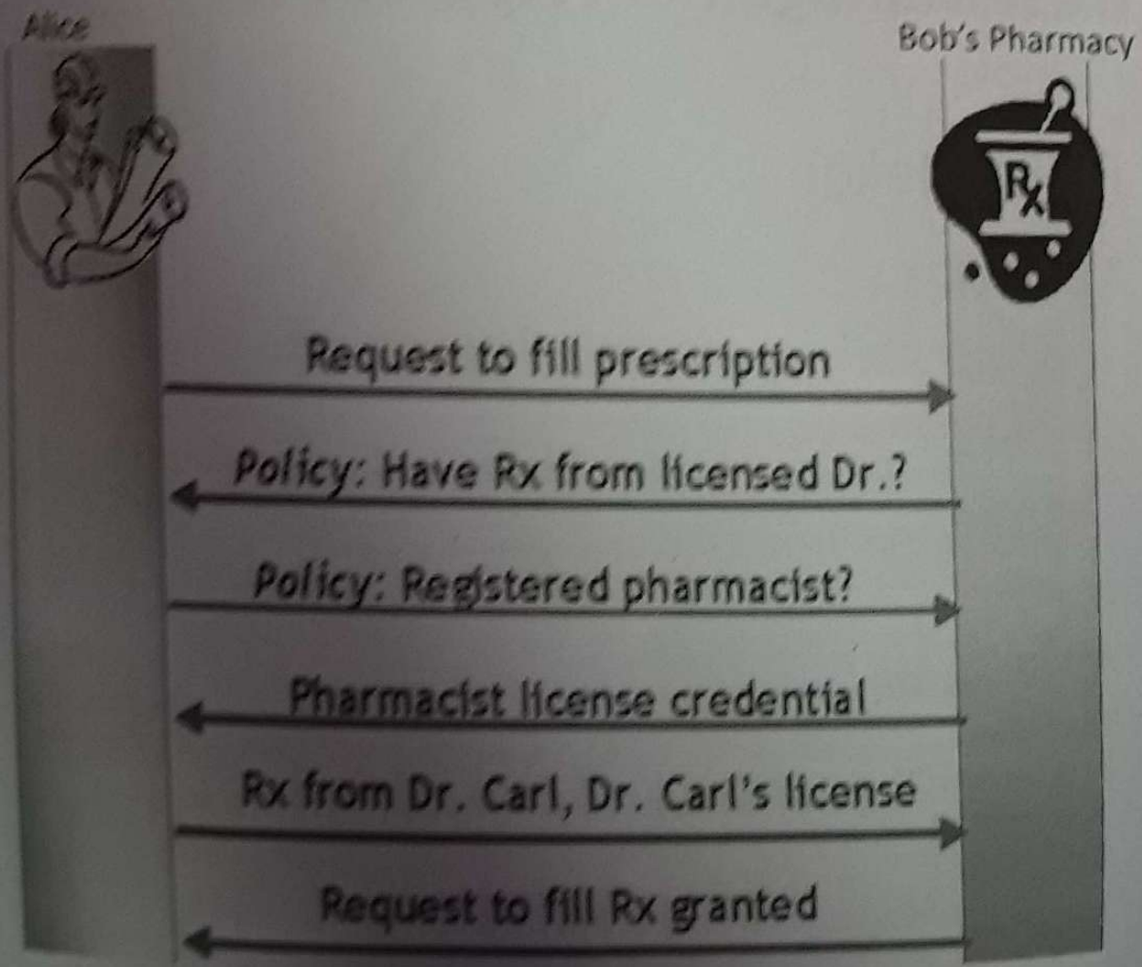
Trust Negotiation

- An automated approach to establish bilateral trust between two parties at run time.

- Gradually disclosure of information by both parties to establish the trust being guided by information disclosure policy of both parties.

- A **process of credential exchange** in which both parties seek to enable a positive authorization decision for the main resource requester, while also supporting the additional authorization decisions that may be necessary to achieve this.

Trust Negotiation



Language for TM System

Datalog

- a language used extensively in deductive database systems.
- Authorization decisions are obtained by evaluating a query involving the client and the requested resource.
- Evaluation in general requires collecting data and rules from distributed repositories.

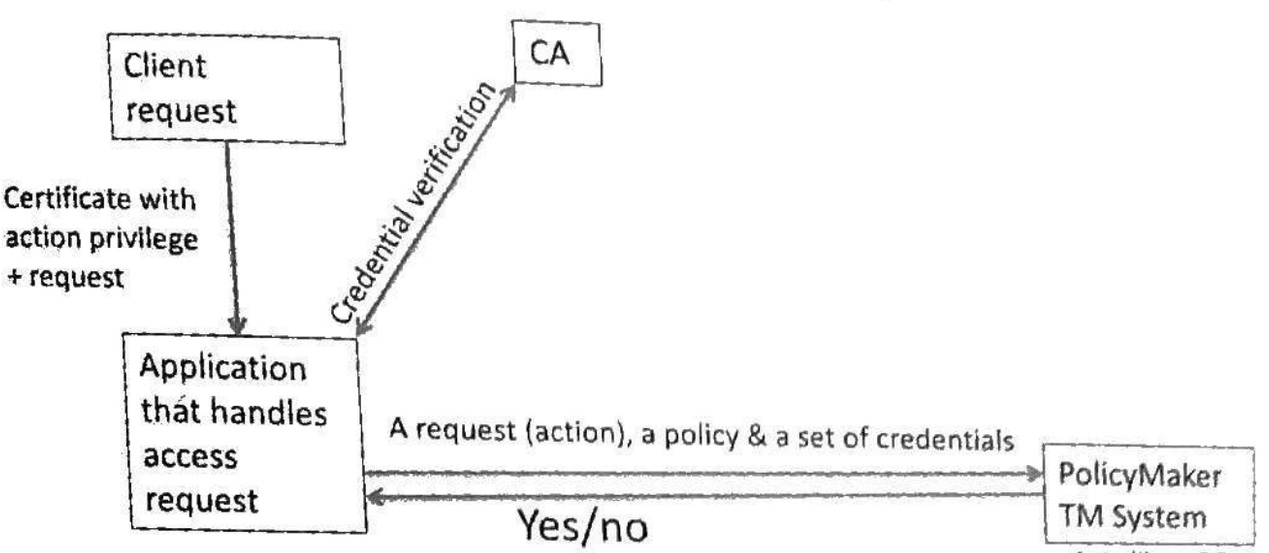
Trust Management Approaches

- PolicyMaker
- KeyNote

PolicyMaker – First TM System

- **General principles of PolicyMaker TM System**
- **Unified mechanism:** Policies, credentials and trust relationships are expressed as programs (or part of programs) in a safe programming language.
- **Flexibility:** can support the complex trust relationship that can occur in the very large network applications.
- **Locality of control:** Each party can decide in each circumstance whether to accept the credentials presented by the second party.
- **Separation of mechanism from policy:** The mechanism for verifying credentials does not depend on the credentials themselves.

PolicyMaker – First TM System



Tasks:
(i) Local policy update,
(ii) Deciding whether additional credential is required
(iii) Credential verification, revocation and
(iv) Sending and receiving trust message.

Tasks:
(i) Input verification and
(ii) sending yes/no

PolicyMaker – First TM System

- An Application plus Trust Management Engine(TME)
- The PolicyMaker service appears to Applications very much like a database query engine.
- TME accepts as input a set of local policy statement, a collection of credentials and a string describing a proposed trusted action.
- It evaluates proposed actions by interpreting the policy statements and credentials.
- It returns either a simple yes/no answer or additional restrictions that would make the proposed action acceptable.

PolicyMaker – First TM System

- The certificates and certificate revocation are obtained by Application. Key verification is also done by Apps.
- Security policies and credentials are defined in terms of predicates (filters) that are associated with public keys.
- Filters accept or reject action descriptions based on what the holders of the corresponding secret keys are trusted to do.
- Security policies and credentials consist of a binding between a filter and one or more public keys.
- Filters can be written in a variety of interpreted languages.

policyMaker – First TM System

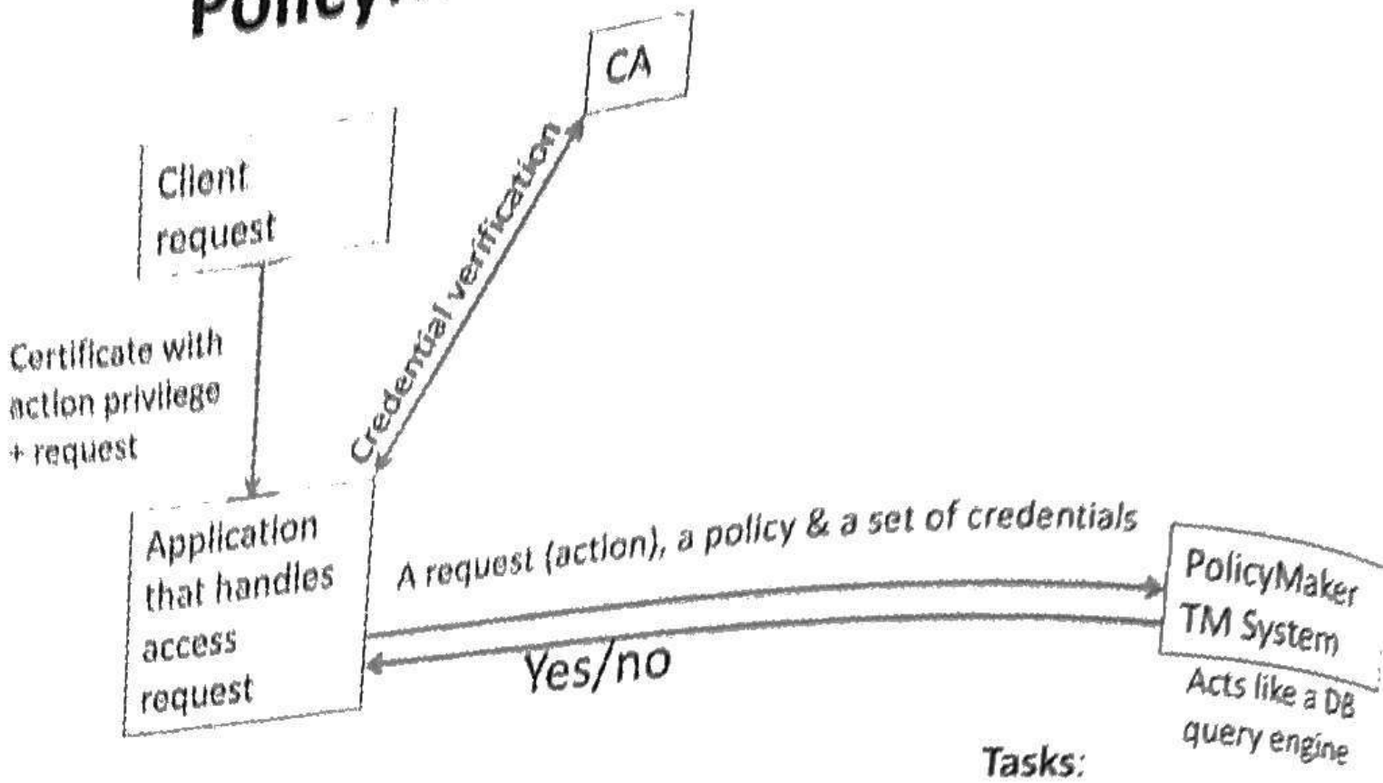
- **Assertions (Security Policies & Credentials) have the form**
  - "<Source> **ASSERTS** <AuthorityStruct> **WHERE** <Filter>;""
  - <Source> field identifies the authority that makes this assertions (CA).
  - <AuthorityStruct> field contains the subjects to whom this assertion applies (Requester).
  - <Filter> field has an Application Specific string <action-string> that must be satisfied for the assertion to hold.
  - The whole assertion states that the <source> trusts the subject<AuthorityStruct> to be associated with <action-string>.

PolicyMaker – First TM System

Example (All employees of all banks):

- A loan request is submitted to an electronic banking system.
- The request may contain EmployeeID, Bank Code, Employee name, Designation and amount requested along with other information.
- **Trust Relationship:**
- Each bank creates a digital certificate for each of its employee.
- The certificate contains EmployeeID, Bank Code & Employee name, Designation.
- **Policy that can handle such request:**
- Two approvals by two employees are required for loans less than $5000
- Three approvals are required for loans >$5000 and <$10000
- Loans >$10000 is not handled online.

PolicyMaker – First TM System



PolicyMaker – First TM System

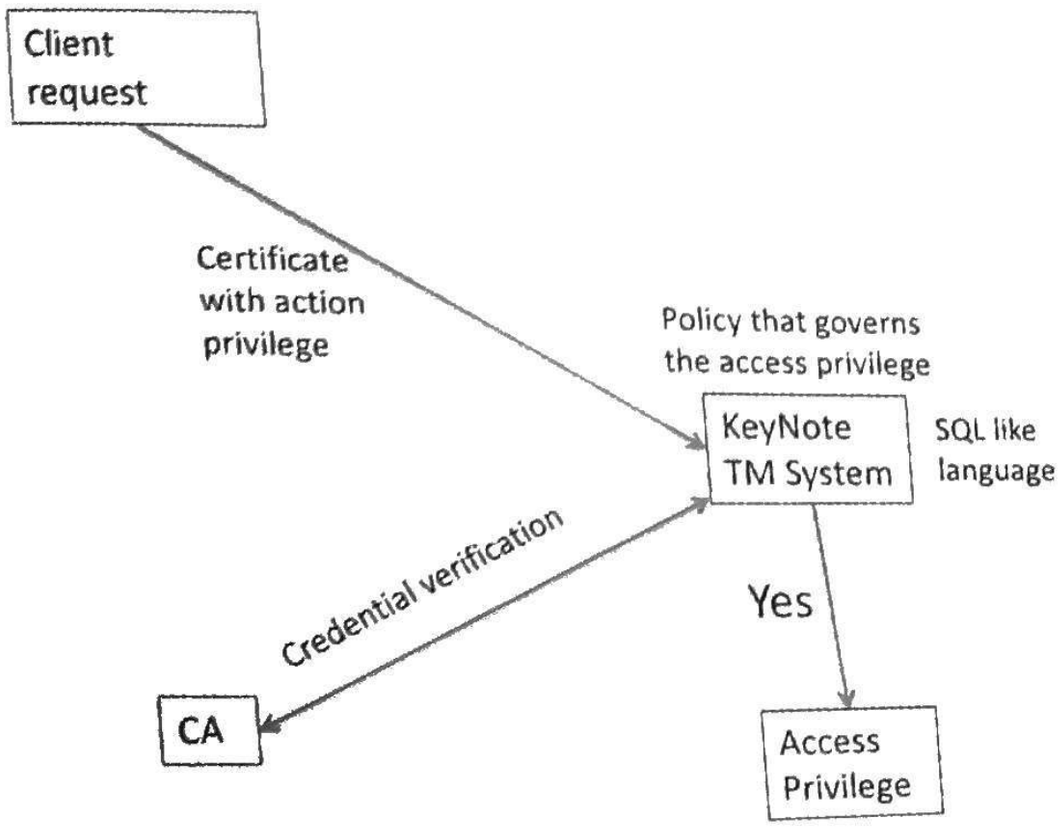
Example (All employees of all banks):

- Each bank each employee to sanction a certain amount of loan the employee wants to borrow from any bank in Bangladesh.
- A loan request is submitted to an electronic banking system.
- The request may contain the name of the requester and amount requested along with other information.
- **Policy that can handle such request:**
- Two approvals are required for loans less than $5000
- Three approvals are required for loans >$5000 and <$10000
- Loans >$10000 is not handled online.
- **Trust Relationship:**
- The head of the loan division must authorize approvers' public key.

KeyNote – TM System

- Direct descendent of PolicyMaker
- KeyNote assertions are written in a specific, concise and human readable assertion language.
- Takes cryptographic signature verification
- Reduces the workload of the calling Applications and better enforces the security policy.
- Relatively a complete software solution for authorization.

KeyNote TM System



Database & Storage Security

Dr. Mohammad Abu Yousuf
yousuf@juniv.edu

Managing and Querying Encrypted Data

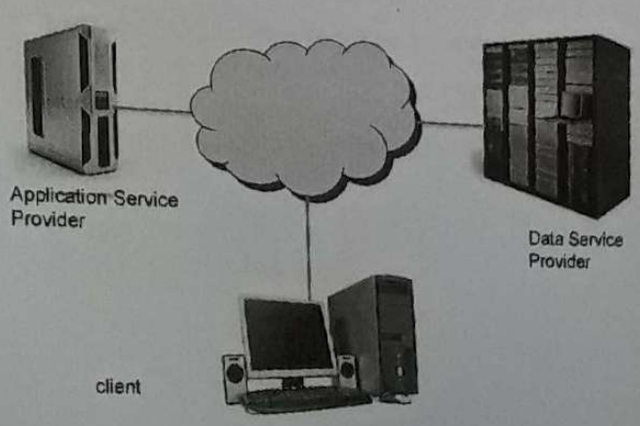
#Lecture-4

Introduction

- Encryption ensures confidentiality
- It has overhead
  - Selection of appropriate algorithm
  - Key management
  - Query on encrypted data
    - Comparison operation
    - Arithmetic operation
    - Keyword matching over encrypted data.
- Only the encryption techniques can not work well.

Introduction

Data As a Service (DAS) model:



Introduction

Data As a Service (DAS) model:

- Data owned by a client is hosted on a third-party server.
- Third Party Service Provider maintains the availability of the system.
- Client will have access to the data as and when required.
- Client will only use the system on rental basis.

Introduction

Data As a Service (DAS) model:

- Confidentiality for sensitive data to be ensured.
- Client hides the sensitive portion of the data at the server side.
- Server side DBA will not get access to these sensitive & confidential data in plaintext form.

Introduction

Two new challenges emerge:

- Efficient encryption algorithms for relational data
- Supporting query on the encrypted relational data.

Introduction

- An application that has driven a lot of research in the cryptographic community is that of keyword-matching over encrypted text data.
- Example : Secure email Service
  - Email server stores emails of Account holders in encrypted format.
  - It allows users to search emails based on keywords without having to decrypt the documents on the server.
  - The returned email then can be decrypted at the client's machine.

Introduction

Examples: Secure Personal Storage Service

- Applications have been developed to let individuals store a variety of data on remote servers and
- Access them over the network securely from any place.

Introduction

Examples: Secure Personal Storage Service

- pVault
  - stores and manages an individuals passwords for his online accounts.
  - Supports password generation and mobile access.
- DataVault
  - works as a secure network drive available online for individuals
  - Google drive

Introduction

Applications for securing XML data

- In case of XML, not only the data but structure is also important, which brings up new kinds of challenges.
- XML text and data travel over the insecure Internet.
- As a result, both Data and structure should be protected.

All these bring a new challenge

Encrypted Data Management

Key concern:

- Confidentiality of the sensitive information in the DB residing on the server.
  - From outside hackers
  - From malicious insiders (DBA).

Solution:

- Encrypt the sensitive portions of data where only the client has access to the key.

Encrypted Data Management

Issues related to encrypted data management:

- Support for encryption algorithm
- Key management
- Query execution on encrypted data

Architecture of a typical DAS system: DAS - Storing and querying encrypted data

- The DAS model offers a variety of data management functionalities in the form of service to clients.
- A key concern in such an application is that of confidentiality of the sensitive information in the database residing on the server.
- In many cases, some or all of the data might be considered sensitive and needs to be protected from any kind of unauthorized access on the server side.
- "Unauthorized access" could refer to a break-in by hackers or an access by a legitimate, but malicious insider, for example a database administrator.

Architecture of a typical DAS system: DAS - Storing and querying encrypted data

- DAS set up and security model
- Querying encrypted relational data
- Relational encryption and storage model
- Keyword search on encrypted text data
- Search over encrypted XML data

DAS setup and security model

- Data-owner
  - One or more clients.
- Server: owner stores the data on the server.
- Clients may query/modify parts of the data remotely according to their access rights.
- Data must be encrypted on the server and only decrypted on the client-side.

DAS setup and security model

- In DAS Application, the client/owner side environment is assumed to be secure and trusted.
- Main problem from server side adversaries
  - A malicious DBA can harm the data owner.
  - Sensitive portion of the data must be in encrypted format on the server at all the time.
  - Secret encryption key should remain with the client.
  - Data is only decrypted on the client side.

Querying Encrypted Relational Data

Suppose an user (Alice) is outsourcing the following tables:

- EMP(eid, ename, salary, addr, did)
- DEPARTMENT(did, dname, mgr)

✓ These tables are stored at the service provider.

✓ Since the service provider is untrusted, the relations must be stored in an encrypted form.

✓ It is assumed that each row is encrypted as a single unit.

✓ Thus we have a set of encrypted records at the server.

Querying Encrypted Relational Data

Client's (Alice) query execution:

SELECT SUM(E.salary)
FROM EMP as E, DEPARTMENT D
WHERE E.did=D.did AND D.mgr="Bob":

The goal in DAS is to process the query directly at the server without the need to decrypt the data.

Querying Encrypted Relational Data

- An approach Alice could use to evaluate such a query might be to request the server for the encrypted form of the *EMP and DEPARTMENT tables.*
- The client could then decrypt the tables and execute the query. This however, would defeat the purpose of database outsourcing, reducing it to essentially a remote secure storage.
- Instead, the goal in DAS is to process the queries directly at the server without the need to decrypt the data.

Querying Encrypted Relational Data

- Basic operator in query language:
  - Comparison operators
    - (=, ≠, <, ≤, >, ≥)
    - Compare attribute values with of a given record with constants
      - SELECT sal > 45000 as in select query
  - Arithmetic operators
    - (+, -, x, /)

Querying Encrypted Relational Data

- SELECT SUM (E.salary)
  from EMP as E, DEPT as D
  where E.did = D.did AND D.mgr = 'Smith'
- Illustrates both classes of operators

Mechanism to Support the Operators

First challenge:

- To develop mechanisms to support comparison and arithmetic operations on encrypted data.

Two categories:

- Approaches based on new encryption techniques.
- Information-hiding based Approaches.

Approaches based on new encryption techniques

- Support either arithmetic and/or comparison operators
  - As privacy homomorphism) supports both arithmetic operations, and doesn't allow comparison.
  - Order preserving encryption supports comparison or selection, sorting, grouping, but cannot support arithmetic operation on server site.
    - Can't support operation on server site
- The limitation:
  - Only safe under limited situations where the adversary knowledge is limited.

Information-hiding based Approaches

Stores additional auxiliary information along with encrypted data to facilitate query evaluation at the server side.

- It may reveal partial information about the data to the server.
- Such auxiliary information are stored in the server in the form of secure indices.
- Secure indices are designed carefully exploiting information hiding mechanism .

Information-hiding based Approaches

- Three techniques (Disclosure Control Methods)
  - Pertubation : For a numeric attribute of a record, add a random value to the true value (numeric attribute).
  - Generalization: Replace a numeric or categorical value by a more general value
  - Swapping: swap the values of a specific attribute of two records

Difference between Disclosure Control method and Cryptographic approach

- In cryptographic approach, if it is broken (disclosure of key), whole information is disclosed.
- In disclosure control method, there may partial disclosure of information or probabilistic in nature.

Difference between Disclosure Control method and Cryptographic approach

- Cryptographic approach works well for arithmetic operations.
- Information hiding approach works well comparison operation.
- However, information hiding approach can not exactly identify the aggregation target group

Query Processing Architecture for DAS

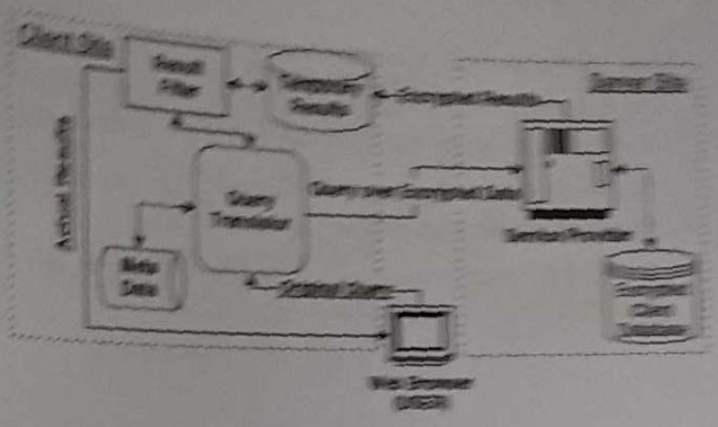


Fig. 1. Query Processing in DAS

Query Processing Architecture for DAS

- Information hiding technique is used.
- Three entities: user, client and server.
- The client stores the data at the server which is managed by the service provider.
- Data is stored in encrypted format in the server most of the time.
- The encrypted database is augmented with some indexing information(secure indices) that allow certain types of query processing at the server level.

Query Processing Architecture for DAS

- The clients also maintains a metadata for
  - translating user queries to the appropriate representation on the server and
  - performs post-processing on server query results.

Query Processing Architecture for DAS

- Based on the auxiliary information used to query over un-encrypted relations are broken in
  - A server-query over encrypted relations which runs on the server and
  - A client-query which runs on the client and processes the results returned after each server query.