University of Trento

Advanced Programming of Cryptographic Methods

Outsourced Sensitive Database

Filippo De Grandi

Date: 20/11/2025

Contents

Description

In this scenario, a client with limited computational resources (such as a smartphone or IoT device) wishes to outsource sensitive documents to a cloud server. The client must later search over these outsourced documents without revealing the content of either the data or the search queries.

The server is honest-but-curious: it follows protocol but attempts to infer as much information as possible. The system must support secure document uploads, updates, deletions, and keyword searches while preventing the server from learning search patterns, access patterns, or update patterns.

Strong privacy guarantees (including forward and backward privacy) and efficient handling of encrypted data are required despite the client's limited capabilities.

Requirements

Security Requirements

In this section, we outline the security requirements for the outsourced sensitive database system, focusing on confidentiality, privacy, integrity, and threat mitigation.

Confidentialiy

> SR 1 **Data Confidentiality** : All outsourced documents must remain encrypted at all times; the server must not be able to read document contents.

> SR 2 **Query Confidentiality** : The content of search queries (keywords) must remain hidden from the server.

> SR 3 **Index Confidentiality** : Metadata such as keyword-document relations must also be encrypted or obfuscated.

Forward & Backward Privacy

> SR 4 **Forward Privacy** : Newly added documents must not be linkable to past queries, i.e., after a keyword has been searched, the server cannot infer that a newly added document contains that keyword.

> SR 5 **Backward Privacy** : After deleting a document, the server should not be able to return it in future searches nor infer previously associated keywords.

Leakage Minimization

> SR 6 **Search Pattern Privacy** : The server should not be able to determine whether two search tokens correspond to the same keyword.

> SR 7 **Access Pattern Privacy** : The server should not learn which documents match a query.

> SR 8 **Update Pattern Privacy** : The server must not be able to link updates (new documents, deletions) to past queries.

Integrity & Freshness

SR 9 **Integrity of Search Results** : The server must not omit or alter returned encrypted documents. The client must be able to verify correctness (e.g. via MACs or authenticated data structures).

SR 10 **Index Integrity** : Any tampering with the encrypted index must be detectable.

SR 11 **Freshness Guarantees** : Returned results must reflect the most recent updates (no replay of outdated search results).

Authentication & Access Control

SR 12 **Authorized Client Access** : Only the legitimate client may upload documents, perform searches, or request updates.

SR 13 **Secure Client–Server Communication** : All communication must use secure channels, despite encryption of data at rest.

Threat Mitigation

SR 14 **Resistance to Traffic Analysis** : The system should minimize information leakage through message size, timing, or frequency.

SR 15 **Compromise Resilience** : If the client device is compromised, the attacker should not be able to derive the plaintext or reveal past queries (e.g. key rotation or forward-secure key updates).

Functional Requirements

In this section, we outline the functional requirements for the outsourced sensitive database system.

FR 1 **Document Upload** : The client must be able to upload encrypted documents to the server

FR 2 **Document Update** : The client must be able to add new encrypted documents without reuploading or re-encrypting the entire dataset.

FR 3 **Document Deletion** : The client must be able to delete previously outsourced documents in a way that prevents future retrieval.

FR 4 **Search Capability** : The client must be able to issue search queries for specific keywords.

FR 5 **Result Retrieval** : The system must return a set of encrypted documents relevant to the search query.

FR 6 **Efficient Indexing** : The server must maintain an encrypted index that supports efficient search and update operations.

FR 7 **Lightweight Client Operations** : Most computation (e.g. index management, filtering) should occur server-side because the client device is resource constrained.

Non-Functional Requirements

Performance

NFR 1 **Low Client Overhead** : Client operations must be computationally lightweight due to limited resources.

NFR 2 **Efficient Search** : Search latency must remain practical even for large datasets.

NFR 3 **Efficient Updates** : Adding or deleting documents should not require rebuilding entire encrypted indexes.

Scalability

NFR 4 **Scalable Storage** : The solution must support large-scale datasets.

NFR 5 **Distributed Deployment** : The server-side components should support distributed cloud infrastructure without breaking security guarantees.

Usability

NFR 6 **Seamless Client Experience** : Search and update operations should appear seamless to the client user.

NFR 7 **Minimal Management** : The system should automate as much of the cryptographic key management as possible.

Reliability & Availability

NFR 8 **High Availability** : The cloud service must maintain uptime for search and update requests.

NFR 9 **Resilience to Server Failure** : Encrypted data should survive server migrations or failures without requiring re-encryption by the client.

Compliance & Governance

NFR 10 **GDPR-compliant Deletion** : Deletion must make documents irrecoverable (cryptographic erasure).

NFR 11 **Audit Logging** : The system should log access, updates, and cryptographic operations in a privacy-preserving format.

NFR 12 **Regulatory Compliance** : The architecture must comply with data protection regulations depending on deployment (HIPAA, GDPR, etc.).