University of Trento

Advanced Programming of Cryptographic Methods

Outsourced Sensitive Database

Filippo De Grandi

Date: 27/11/2025

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

Functional Requirements

SR 1 **Add documents** : The user must be able to add documents to the database

SR 2 **Delete documents** : The user must be able to delete documents from the database

SR 3 **Searching** : The database must return the documents related to the key words searched by the client.

SR 4 **User search** : The user must be able to search using key words.

SR 5 **Document keyword association** : The documents must be associated with specific keywords.

Security Requirements

FR 1 **Non-revealing database** : The database must not leak sensitive information to the server or third-parties.

FR 2 **User scope** : The user must be able to obtain only their personal documents.

FR 3 **Data integrity** : The integrity of the information must be kept both during communication and inside the database.

FR 4 **Authentication** : The users must be authenticated before communication and for sensitive operations.

FR 5 **Secure queries** : The queries must reveal as little information as possible

Non-Functional Requirements

NFR 1 **Compliance** : The system must comply with local regulations for sensitive data.

Technical Details

Architecture

The system follows a client-server model optimized for secure outsourced storage and search functionalities. The client (resource-limited device) performs all key-related and sensitive operations, while the cloud server stores encrypted data and executes search operations without learning their content.

Components

1. Client Module:
   - Generates and stores cryptographic keys.
   - Encrypts documents before uploading.
   - Creates encrypted search tokens (trapdoors) for querying.
   - Verifies search results returned by the server.
2. Server Module:
   - Stores encrypted documents and an encrypted index.
   - Processes search queries using trapdoors.
   - Returns encrypted search results to the client.
   - Maintains access logs for auditing purposes.
3. Cryptographic Module:
   - *Dynamic Searchable Symmetric Encryption* (DSSE) [1] scheme for secure document storage and search.
   - Forward-secure key update mechanism for forward privacy.
   - Authenticated data structure for integrity verification.
   - Secure communication protocols (TLS) for data in transit.

Workflow

1. Upload: Client encrypts a document + keyword associations ⟶ sends ciphertext + encrypted index entries.
2. Search: Client sends trapdoor for a keyword ⟶ server performs encrypted lookup ⟶ returns encrypted matches.
3. Deletion: Client sends a deletion token referencing encrypted index entries ⟶ server removes them.
4. Integrity Check: Returned results include MACs or path proofs which the client verifies.

Security Considerations

Confidentiality & Privacy

- DSSE ensures the server learns only minimal leakage.
- Forward privacy: New document additions cannot be linked to past queries because trapdoor keys evolve.
- Backward privacy: Deleted documents cannot be returned in future searches.
- Encrypted index: Keyword–document associations remain hidden.

Integrity & Authenticity

- MACs or Merkle proofs ensure the server cannot forge or alter documents.
- Authenticated channels (TLS + client keys) enforce secure communication.
- The client verifies integrity of results, protecting against server tampering.

Access Control

- Each client uses a unique key. The server cannot decrypt documents or impersonate users.
- Only the legitimate client can generate valid search tokens or upload/delete entries.

Leaks Reduction

The system mitigates:

- Search pattern leakage using probabilistic trapdoors.
- Access pattern leakage via randomized structures.
- Update pattern leakage by unlinking update tokens.

Threat Model

- The server is honest-but-curious as it follows protocol but tries to infer patterns.
- Even if the server stores all interactions, search tokens remain unlinkable due to the evolving trapdoor function.
- Network attackers are mitigated via TLS and standard authenticated encryption.

Bibliography

[1] Wikipedia, “Searchable Symmetric Encryption.” [Online]. Available: https://en.wikipedia.org/wiki/Searchable_symmetric_encryption