P2P-Messaging App Report – Phase 1

Henrique Marques (57153), Gonçalo Fernandes (58194), Miguel Pena (64446)

This report analyzes the features implemented in phase 2 for the P2P Messaging System developed for the class of Data Privacy and Security in the java programming language. In phase 2, the tasks chosen were: long term message storage and message searching.

Tasks:

- **Long-term message storage**: For this task we decided to implement our own servers, this way we were able to have control over all details and simplify the implementation process without the need to use a REST API. To achieve long-term storage we used secret sharing, opting to implement the Shamir Secret Sharing Scheme.
- **Message searching**: For this task we also decided to use the servers already created to enable message searching. To achieve message searching we decided to implement Cash Searchable Encryption.

Solution overview:

- **Long-term storage:** The solution for long-term message storage utilizes a combination of encryption, secret sharing, and fault tolerance. To manage the storage and retrieval of message data across multiple servers, we implemented a method called *trySendToServers*. This method handles the process of encrypting the messages, splitting them into shares, and distributing these shares to backup servers using SSL sockets. Each share sent to the servers is encrypted and only the client that sent them can decode them. In the event of server failures, the system is designed to fall back to local backups, ensuring data integrity and availability. Once connection to at least the necessary servers can be established each server will receive a share of the updated file. Shares are sent to the backup servers only after a new message has been added to the file, and once a connection to the required number of servers is successfully established (the number of servers required is determined by the share threshold necessary to recreate the file).

- **Message searching:** The solution for message searching utilizes a combination of encryption, searchable encryption, and fault tolerance. To enable message searching across multiple servers, we implemented a method called *searchInConversations*. This method handles the process of encrypting the search term and connecting to one of the available servers (we assume all servers have the same information although this might not be the case) using SSL sockets. The server then uses this term to search all entries that it has. It then returns the search results which the client has to decode. Search terms are sent to the servers using the *updateSearchTerms* method whenever a message is sent to another client.

Security Analysis - Security Guarantees Offered

The following security guarantees are provided by **Shamir Secret Sharing Scheme**:

Confidentiality: The security of Shamir's Secret Sharing relies on the properties of polynomial interpolation. Without enough shares, no participant or adversary can learn anything about the secret. This is a strong form of security that does not rely on the computational difficulty of certain problems, but rather on the inherent properties of the scheme.

Minimal information leakage: In Shamir's Secret Sharing, only the minimum number of shares required, *t*, can reveal the secret. Any set of fewer than *t* shares reveals no useful information. This means the secret is highly secure unless an adversary collects enough shares. And in our case each share is encrypted so only the client with the correct keys can decode them.

No Single Point of Failure: The secret is distributed across multiple participants, reducing the risk associated with a single point of failure. If one or more participants are compromised or malicious, the secret can still remain secure as long as the threshold number of shares are protected.

The following security guarantees are provided by **Cash Searchable Encryption Scheme**:

Data Confidentiality: The primary guarantee of CashSE is the confidentiality of the underlying plaintext data. Even though the data is stored on an untrusted server, encryption ensures that unauthorized entities, including the server itself, cannot access or infer the actual content of the stored data. This is particularly important because the data stored on the servers corresponds to personal communications between peers.

Query Privacy: CashSE protects the privacy of search queries by ensuring that the server cannot learn the content of the client's query. This is achieved by encrypting the search query before it is sent to the server, effectively obfuscating its meaning. As a result, the server performs the search operation without knowing what the client is searching for, preventing potential profiling or misuse of search information.

Result Privacy: Ensures that only the client has access to the outcome of a search operation. While the server facilitates the search by identifying matching records, it does not learn which specific records are returned or their contents. This protects both the data being queried and the results retrieved from exposure.

Limitations and Considerations

- **Long-term storage:** A key issue is the potential exposure of shares. If an adversary gains access to enough shares, they can reconstruct the secret. Additionally, the loss of a share by a participant can make the secret unrecoverable if the threshold is not met. The scheme also relies on secure management and storage of shares, making it vulnerable to physical or digital breaches if shares are mishandled.

  Another limitation is that the threshold must be carefully chosen; too low and the risk of the secret being exposed increases, while too high a threshold can make reconstruction difficult. In this case the threshold might be too low because of the number of available servers, although that number could be scaled up (not necessary for demonstration purposes). The scheme is also prone to vulnerabilities from collusion among participants, although in this case each share is also encrypted using hybrid encryption. Furthermore, Shamir's Secret Sharing lacks integrity checks, meaning there's no built-in mechanism to verify share authenticity or prevent tampering. Lastly, this scheme also does not offer a way to recover shares.

- **Message searching**: While offering strong confidentiality and query privacy, the scheme used faces significant limitations. One major drawback is pattern leakage, where the server can observe search patterns and access patterns. This information can be exploited to infer sensitive details, particularly when combined with statistical analysis or prior knowledge. Additionally in this case, Cash Searchable Encryption supports only keyword searches, limiting the complexity of queries. Performance overhead is another concern, as encryption and search processes incur higher computational costs compared to plaintext operations, and dynamic updates often require re-encrypting data, adding further inefficiencies.
  This scheme is also vulnerable to statistical inference attacks that can exploit leaked patterns or keyword frequency distributions to deduce encrypted query contents or document contents. Chosen keyword attacks allow adversaries to infer query details by injecting specific keywords into the dataset and observing search results.