Searchable Symmetric Encryption: Improved Definitions and Efficient Constructions

Saptarshi Gan

Indian Institute of Technology, Kanpur

sapgan@cse.iitk.ac.in

October 26, 2014
Overview

1. Symmetric Encryption
   - Definition
   - Example

2. Searchable Symmetric Encryption

3. Previous Works

4. Limitations

5. New Schemes
Definition (Symmetric Encryption)

Symmetric-key algorithms are a class of algorithms for cryptography that use the same cryptographic keys for both encryption of plaintext and decryption of ciphertext. The keys may be identical or there may be a simple transformation to go between the two keys.
Example

Plain-text input

“The quick brown fox jumps over the lazy dog”

Cipher-text

“AxCv;5bmEseTfid3)fGsmWe#4^,sdgfMwir3:dkJeTsY8R\s@!q3 %”

Plain-text output

“The quick brown fox jumps over the lazy dog”

Encryption

Decryption

Same key (shared secret)
What is searchable symmetric encryption?

Searchable symmetric encryption (SSE) allows a party to outsource the storage of its data to another party (a server) in a private manner, while maintaining the ability to selectively search over it.
SAE vs SSE

- SAE - index added by anyone, trapdoor only by user
- SAE - Not very efficient
Previous Works in SSE

- Major work by Ostrovsky and Goldreich
- Poly-logarithmic overhead for all parameters
- Preprocessing order - size of the data
Limitations in previous schemes

- Leaked search outcome and user search pattern
- Indexed search and trapdoor generation not secure from server
- Non-adaptive
- Used classical definition of security (Server not supposed to search)
Definition (Non-Adaptive Indistinguishability Security for SSE informal version)

A SSE scheme is secure in the sense of non-adaptive indistinguishability if for any two adversarially constructed histories with equal length and trace, no (probabilistic polynomial-time) adversary can distinguish the view of one from the view of the other with probability non-negligibly better than $1/2$. 
## Schemes Provided

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Description</th>
</tr>
</thead>
</table>
| **SSE-1** | - Efficiency similar to previous schemes  
                   - Secure according to new definitions                                |
| **SSE-2** | - More server storage used  
                   - Higher communication size per query  
                   - Secure against adaptive adversary  
                   - Asymptotic cost similar to SSE-1                                      |
| **Multi-SSE** | - Uses SSE with BE (Broadcast Encryption)  
                   - Multiple users can search documents  
                   - Group with access right is dynamic (for BE)                           |
Comparision of different schemes

<table>
<thead>
<tr>
<th>Properties</th>
<th>[24, 18]</th>
<th>[24, 18]-light</th>
<th>[27]</th>
<th>[16]</th>
<th>[12]</th>
<th>SSE-1</th>
<th>SSE-2</th>
</tr>
</thead>
<tbody>
<tr>
<td>hides access pattern</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>server computation</td>
<td>$O(\log^3 n)$</td>
<td>$O(\sqrt{n})$</td>
<td>$O(n)$</td>
<td>$O(n)$</td>
<td>$O(n)$</td>
<td>$O(1)$</td>
<td>$O(1)$</td>
</tr>
<tr>
<td>server storage</td>
<td>$O(n \cdot \log n)$</td>
<td>$O(n)$</td>
<td>$O(n)$</td>
<td>$O(n)$</td>
<td>$O(n)$</td>
<td>$O(n)$</td>
<td>$O(n)$</td>
</tr>
<tr>
<td>number of rounds</td>
<td>$\log n$</td>
<td>$2$</td>
<td>$1$</td>
<td>$1$</td>
<td>$1$</td>
<td>$1$</td>
<td>$1$</td>
</tr>
<tr>
<td>communication</td>
<td>$O(\log^3 n)$</td>
<td>$O(\sqrt{n})$</td>
<td>$O(1)$</td>
<td>$O(1)$</td>
<td>$O(1)$</td>
<td>$O(1)$</td>
<td>$O(1)$</td>
</tr>
<tr>
<td>adaptive adversaries</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>yes</td>
</tr>
</tbody>
</table>

**Figure**: Properties and performance (per query) of various SSE schemes. $n$ denotes the number of documents in the document collection. For communication costs, we consider only the overhead and omit the size of the retrieved documents, which is the same for all schemes. For server computation, we show the costs per returned document. For simplicity, the security parameter is not included as a factor for the relevant costs.
The End