Introduction to Mylar
A visual guide
Threat Model

Client's Browser

Application Server

Application Code

DB
Threat Model

*Assume site owner/developer not malicious (will not leak keys)
Threat Model

Client's Browser → Application Code → Encryption / Decryption Layer → DB

Application Server
Mylar’s Approach

Client’s Browser

Encryption / Decryption Layer

Application Server

Application Code

DB
Mylar’s Approach

Client’s Browser

Application Server

Application Code

DB

Encryption / Decryption Layer
Mylar’s Approach

1. Ensure Code Integrity?

2. Share Data Privately?

3. Computation (Keyword Search)?
Mylar’s Approach

1. Ensure Code Integrity?

Client's Browser
   Encryption / Decryption Layer

Client 2's Browser
   Encryption / Decryption Layer

Application Server

Application Code

DB

2. Share Data Privately?

3. Computation (Keyword Search)?
Issue 1 - Ensuring Code Integrity

Primary Origin

X.509 Certificate with mylar_pubkey

Browser Extension

mylar_hash parameter

Second Origin

```html
<html>
<head>
  <script src="app-logic.js"></script>
</head>
<body>
  <div>LOL</div>
</body>
</html>
```
Mylar’s Approach

1. Ensure Code Integrity?

2. Share Data Privately?

3. Computation (Keyword Search)?
Issue 2 - Share Data Privately

- Represents an application-level access control entity.
- E.g. user, group, shared document

**Principle**

- Name
- Public/Private Key Pair
  - Wrapped Keys
  - Certificate
  - Data
  - Password Encrypted
  - Private Key

- Principle's private key encrypted (wrapped) by other principal's public key
- To verify public key belongs to principal
- Data is encrypted by principal's public key
- Only for User principals
Issue 2 - Share Data Privately

1. Alice generates “Shared Document” pub/priv key pair
2. Create wrapped key $E(\text{Priv}_{\text{Shared Doc}}, \text{Pub}_{\text{Alice}})$
Issue 2 – Share Data Privately

3. Signs and publishes “Shared Document” principal

Client-Side

Server-Side

Have Access To

Alice

Bob

Shared Document
Issue 2 - Share Data Privately

Client-Side

Alice

Bob

Have Access To

Server-Side

Alice

Bob

Shared Document
4. Ask server for Bob public key
5. Creates a wrapped key for Bob
   \[ E(\text{Priv}_{\text{Shared Doc}}, \text{Pub}_{\text{Bob}}) \]
6. Grants access to Bob by sending the wrapped key to the Shared Document principal.
Issue 2 – Share Data Privately

7. \( E(\text{"I Love You"}, \text{Pub}_{\text{Shared Doc}}) \)

8. \( D(\text{Wrapped Key & Encrypted Data}_{\text{Bob}}, \text{Priv}_{\text{Bob}}) = \text{Priv}_{\text{Shared Doc}} \)

9. \( D(\text{Encrypted Data}, \text{Priv}_{\text{Shared Doc}}) = \text{"I Love You"} \)
Issue 2 - Share Data Privately

Access Graph

Alice

Have access to

Shared Document

Have access to

Bob
Issue 2 - Share Data Privately
Issue 2 - Share Data Privately

Certificate Graph

Out of control of attacker

IDP

Static Principals

Alice

Bob

Canary

Damien

Shared Group

Shared Group

Cluster 1

Cluster 2

Shared Doc
Mylar’s Approach

1. Ensure Code Integrity?
2. Share Data Privately?
3. Computation (Keyword Search)?
Issue 3 – Computation Over Encrypted Data (Search)

Document A Principal

\[ H(“Apple”)^{Key_A} = “tyu32hj4” \]

Document B Principal

\[ H(“Apple”)^{Key_B} = “387nmhns” \]

\[ H_2 (r, e(H(\text{Word}), g)^{Key}) \]

Nice Property: \( e(H(w)^a, g^b) = e(H(w), g)^{ab} \)
Issue 3 - Computation Over Encrypted Data (Search)

\[ h' = H_2(r, atk) \]
\[ = H_2(r, e(tk, \Delta_{KeyA \rightarrow KeyB})) \]
\[ = H_2(r, e(H("Apple")^{KeyA}, \Delta_{KeyA \rightarrow KeyB})) \]
\[ = H_2(r, e(H("Apple"), g^{KeyB/KeyA})) \]
\[ = H_2(r, e(H("Apple"), g^{KeyB})) \]

\[ h = H_2(r, e(H("Apple"), g^{KeyB})) \]
Guarantees

- Data confidentiality in the face of arbitrary server compromises
  - As long as none of the users that have access to the data is compromised

- Data Authenticity
  - But not freshness or correctness
## How is Mylar Different?

<table>
<thead>
<tr>
<th>Mylar</th>
<th>CryptDB</th>
</tr>
</thead>
<tbody>
<tr>
<td>● Threat model assumes entire server compromised</td>
<td>● 2 Threat Models</td>
</tr>
<tr>
<td>● Provides guarantees for data confidentiality</td>
<td>● Provides partial guarantees for data confidentiality</td>
</tr>
<tr>
<td>● Only Search operation supported</td>
<td>● Most SQL operations supported</td>
</tr>
<tr>
<td>● Built-in ACL controls and data sharing</td>
<td>● Isolation of user’s data (no sharing)</td>
</tr>
<tr>
<td>● Better suited for NoSQL variant DBs</td>
<td>● Better suited for SQL variant DBs</td>
</tr>
<tr>
<td>● Potentially significant effort on client-side</td>
<td>● Hidden from clients</td>
</tr>
</tbody>
</table>
# Effort and Performance

<table>
<thead>
<tr>
<th>Application</th>
<th>LoC before</th>
<th>LoC added for Mylar</th>
<th>Number and types of fields secured</th>
<th>Existed before?</th>
<th>Keyword search on</th>
</tr>
</thead>
<tbody>
<tr>
<td>kChat [23]</td>
<td>793</td>
<td>45</td>
<td>1 field: chat messages</td>
<td>Yes</td>
<td>messages</td>
</tr>
<tr>
<td>endometriosis</td>
<td>3659</td>
<td>28</td>
<td>tens of medical fields: mood, pain, surgery, ...</td>
<td>Yes</td>
<td>N/A</td>
</tr>
<tr>
<td>submit</td>
<td>8410</td>
<td>40</td>
<td>3 fields: grades, homework, feedback</td>
<td>Yes</td>
<td>homework</td>
</tr>
<tr>
<td>photo sharing</td>
<td>610</td>
<td>32</td>
<td>5 fields: photos, thumbnails, captions, ...</td>
<td>Yes</td>
<td>N/A</td>
</tr>
<tr>
<td>forum</td>
<td>912</td>
<td>39</td>
<td>9 fields: posts body, title, creator, user info, ...</td>
<td>No</td>
<td>posts</td>
</tr>
<tr>
<td>calendar</td>
<td>798</td>
<td>30</td>
<td>8 fields: event body, title, date, user info, ...</td>
<td>No</td>
<td>events</td>
</tr>
<tr>
<td>WebAthena [8]</td>
<td>4800</td>
<td>0</td>
<td>N/A: used for code authentication only</td>
<td>Yes</td>
<td>N/A</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Application</th>
<th>Operation for latency</th>
<th>Latency w/o Mylar</th>
<th>Latency with Mylar</th>
<th>Throughput w/o Mylar</th>
<th>Throughput with Mylar</th>
<th>Throughput units</th>
</tr>
</thead>
<tbody>
<tr>
<td>submit</td>
<td>send and read a submission</td>
<td>65 msec</td>
<td>606 msec</td>
<td>723</td>
<td>394</td>
<td>submissions/min</td>
</tr>
<tr>
<td>submit w/o search</td>
<td></td>
<td>70 msec</td>
<td></td>
<td></td>
<td>595</td>
<td></td>
</tr>
<tr>
<td>endometriosis</td>
<td>fill in/read survey</td>
<td>1516 msec</td>
<td>1582 msec</td>
<td>6993</td>
<td>6130</td>
<td>field updates/min</td>
</tr>
</tbody>
</table>

4x Space Overhead for kChat
Likely to be adopted/implemented in the real world?
Thank You
How do digital signatures work?