Towards enabling Secure Web-Based Cloud Services using Client-Side Encryption

Martin Johns, Alexandra Dirksen

Institute for Application Security
Technische Universität Braunschweig

November 9, 2020
Encrypt all the things!...but web apps?
Encrypt all the things!...but web apps?

- End2End encryption gains popularity
Encrypt all the things!...but web apps?

- End2End encryption gains popularity
- Client-side apps uses native encryption
Encrypt all the things!...but web apps?

- End2End encryption gains popularity
- Client-side apps uses native encryption
- Web-apps can’t use encryption → Active JavaScript attacker
Encrypt all the things!...but web apps?

- End2End encryption gains popularity

- Client-side apps uses native encryption

- Web-apps can’t use encryption
  → Active JavaScript attacker

- Encryption via 3rd-party extension not feasible
Web apps under fire!

Cloud operator provides a web-based application for its cloud services.

Database operator stores the user's data on behalf of the cloud operator.

User utilized her web browser to access the application of the cloud operator.
Web apps under fire!

**Cloud operator** provides a web-based application for its cloud services
Web apps under fire!

Cloud operator provides a web-based application for its cloud services

Database operator stores the user’s data on behalf of the cloud operator
Web apps under fire!

Cloud operator provides a web-based application for its cloud services

Database operator stores the user’s data on behalf of the cloud operator

User utilized her web browser to access the application of the cloud operator
Web apps under fire!
Web apps under fire!
Web apps under fire!
Web apps under fire!
Web apps under fire!

Security Objective

The user want to keep her personal data private while still using the cloud operators web-based application.
Encrypt ALL the things!

- Isolation against untrusted JS
- Protection against malicious code injection
- Protection against UI-redressing attacks
- Incremental deployment possible

Solution
Native encryption tools for web devs via new standardized DOM elements
Encrypt ALL the things!

- Isolation against untrusted JS
Encrypt ALL the things!

- Isolation against untrusted JS
- Protection against malicious code injection
Encrypt ALL the things!

- Isolation against untrusted JS
- Protection against malicious code injection
- Protection against UI-redressing attacks
Encrypt ALL the things!

- Isolation against untrusted JS
- Protection against malicious code injection
- Protection against UI-redressing attacks
- Incremental deployment possible
Encrypt ALL the things!

- Isolation against untrusted JS
- Protection against malicious code injection
- Protection against UI-redressing attacks
- Incremental deployment possible

Solution

Native encryption tools for web devs via new standarized DOM elements
CryptoMembranes (CM)
CryptoMembranes (CM)

1. Usage of CM DOM elements directly in HTML
1. Usage of CM DOM elements directly in HTML

2. Each DOM element $\rightarrow$ CM element (e.g. DIV $\rightarrow$ CryptoDIV)
CryptoMembranes (CM)

1. Usage of CM DOM elements directly in HTML

2. Each DOM element → CM element (e.g. DIV → CryptoDIV)

3. Same interface as corresponding DOM element
CryptoMembranes (CM)

1. Usage of CM DOM elements directly in HTML

2. Each DOM element $\rightarrow$ CM element
   (e.g. DIV $\rightarrow$ CryptoDIV)

3. Same interface as corresponding DOM element

4. CM maintains cipher and plain value
CryptoMembranes (CM)

1. Usage of CM DOM elements directly in HTML

2. Each DOM element $\rightarrow$ CM element (e.g. DIV $\rightarrow$ CryptoDIV)

3. Same interface as corresponding DOM element

4. CM maintains cipher and plain value

5. Plain value displayed to browser UI
CryptoMembranes (CM)

1. Usage of CM DOM elements directly in HTML

2. Each DOM element $\rightarrow$ CM element (e.g. DIV $\rightarrow$ CryptoDIV)

3. Same interface as corresponding DOM element

4. CM maintains cipher and plain value

5. Plain value displayed to browser UI

6. Cipher displayed to website JS
CM Details

Untrusted JavaScript

AB34CEA23...

Hello World
CM Details

Untrusted JavaScript

Untrusted JavaScript

Hello World

Hello World

AB34CEA23...

AB34CEA23...

website HTML

website HTML

browser UI

browser UI

DIV

DIV

Crypto DIV

Crypto DIV
CM Architecture: Encryption

Browser

HTML

JavaScript

hello world
CM Architecture: Encryption

Browser

HTML

JavaScript

user key
CM Architecture: Encryption

[Diagram showing the flow from a Browser to an AppServer with encryption.]
CM Architecture: Encryption

Browser → User Key → AppServer → Database
CM Architecture: Decryption

Database -> AppServer -> Browser

user key

HTML
JavaScript

hello world
CM Details: Display

HTML Elements for data output:

1. `<DIV> AB34CEA23...</DIV>`
CM Details: Display

HTML Elements for data output:

```html
1  <DIV> AB34CEA23... </DIV>
```

Corresponding CM element:

```xml
1  <CryptoDIV CMKeyID="123" CMAlgID="OrderPreserving">
2  AB34CEA23...
3  </CryptoDIV>
```
CM Details: Input

HTML Element for data entry:

```html
1   <INPUT Type="text" Name="confinput">
```
CM Details: Input

HTML Element for data entry:

```
1    <INPUT Type="text" Name="coninput">
```

Corresponding CM element:

```
1    <CryptoINPUT Type="text" Name="coninput" CMKeyID="345"
                CMAlgID="Deterministic">
```
Client-Side Programming

```html
<CryptoDIV ID="CM1" CMKeyID="911" CMAlgID "Deterministic">
</CryptoDIV>

<CryptoINPUT ID="CM2" Type="text" name="conf" CMKeyID="911" CMAlgID="Deterministic" onchange="moveData()">

<script>
function moveData()
{
    var cm1 = document.getElementById("CM1");
    var cm2 = document.getElementById("CM2");
    var cValue = cm2.value // cValue is encrypted
    cm1.innerText = cValue;
}

</script>
```
Legacy Browser Support

Extension-based support of CryptoMembranes

1. Identifying all CM elements in HTML

2. Insertion of CMs secure compartments
Extension Membranes: Workflow

intercept HTTP response
Extension Membranes: Workflow

intercept HTTP response

↓

iterate CMs in DOM

↓

next
ExtensionMembranes: Workflow

intercept HTTP response

iterate CMs in DOM

next

CM: display
Extension Membranes: Workflow

- Intercept HTTP response
- Iterate CMs in DOM
- Next

CM: display
CM: input
Extension Membranes: Workflow

intercept HTTP response

iterate CMs in DOM

next

CM: display

replace with corr. DOM element

CM: input

replace with <span> element
**Extension Membranes: Workflow**

- Interception of HTTP response
- Iteration through CMs in DOM
- Next step

For CM: Display:
- Replace with corresponding DOM element
- Insert iFrame

For CM: Input:
- Replace with `<span>` element
Extension Membranes: Workflow

intercept HTTP response
↓
iterate CMs in DOM
↓
next

CM: display
↓
replace with corr. DOM element

insert iFrame
↓
decrypt values if any

CM: input
↓
replace with <span> element
ExtensionMembranes: Workflow

intercept HTTP response

iterate CMs in DOM

next

CM: display
replace with corr. DOM element

CM: input
replace with <span> element

insert iFrame

decrypt values if any
ExtensionMembranes: Workflow

intercept HTTP response

iterate CMs in DOM

next

CM: display

replace with corr. DOM element

insert iFrame

decrypt values if any

finish HTTP response

CM: input

replace with <span> element
Security Assessment

- Isolation properties
- Protection against code injection
- Protection against UI-redressing attacks
- Incremental Deployability
Security Assessment

☑ Isolation properties

☐ Protection against code injection

☐ Protection against UI-redressing attacks

☐ Incremental Deployability
Security Assessment

☑️ Isolation properties

☑️ Protection against code injection

☐ Protection against UI-redressing attacks

☐ Incremental Deployability
Security Assessment

- ✔ Isolation properties
- ✔ Protection against code injection
- ✔ Protection against UI-redressing attacks
- □ Incremental Deployability
Security Assessment & Conclusion

- ✔ Isolation properties
- ✔ Protection against code injection
- ✔ Protection against UI-redressing attacks
- ✔ Incremental deployability
Future Work

- Native browser implementation
- User study on secure input / visual indicator
- Identifying crypto needs (Order preserving? Searchable? Aggregated?)
Future Work

- Native browser implementation
Future Work

- Native browser implementation
- User study on secure input/visual indicator
Future Work

☐ Native browser implementation

☐ User study on secure input/visual indicator

☐ Identifying crypto needs
  (Order preserving? Searchable? Aggregated?)