RUB

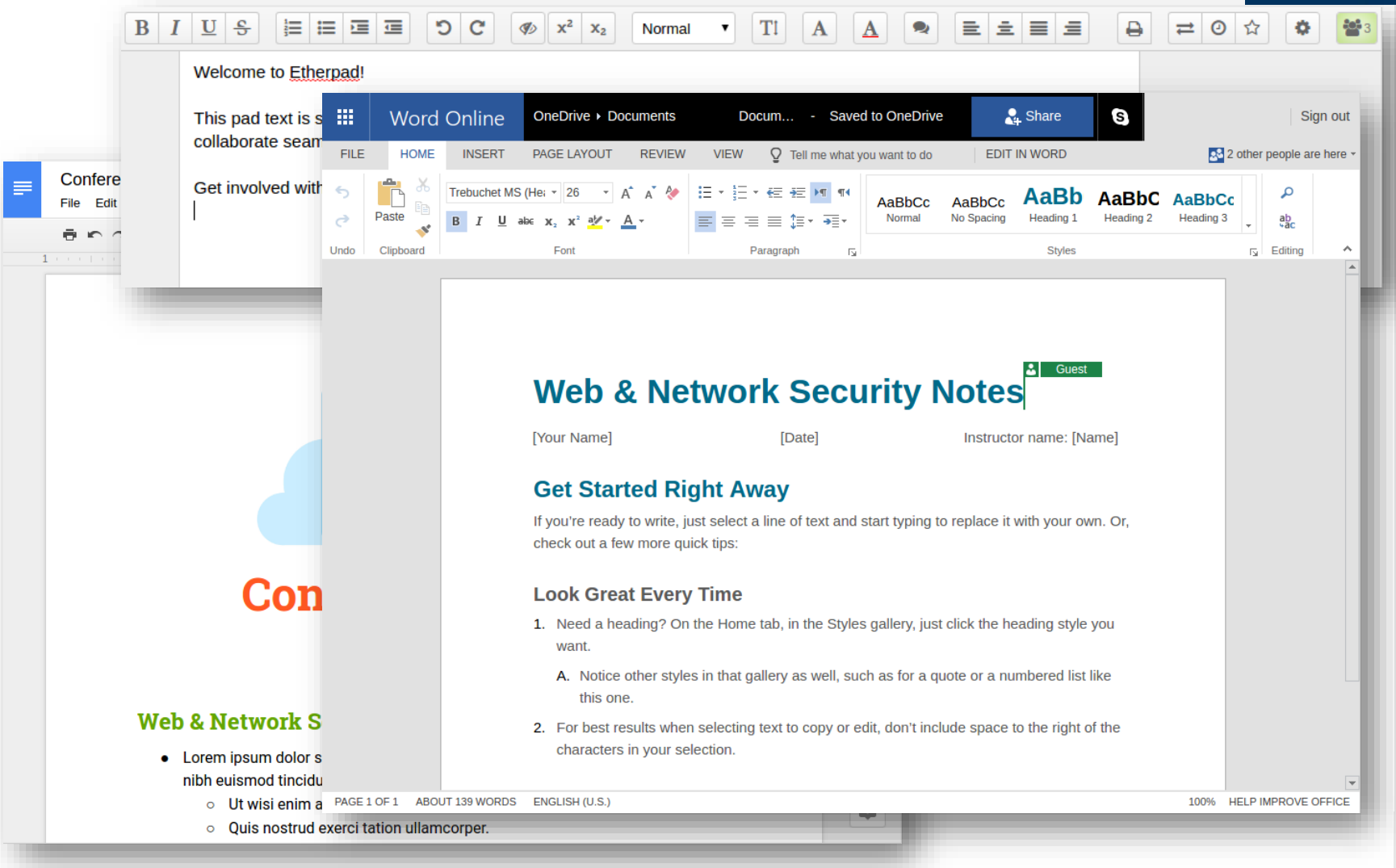
RUHR-UNIVERSITÄT BOCHUM

SECRET: On the Feasibility of a Secure, Efficient, and Collaborative Real-Time Web Editor

Dennis Felsch, Christian Mainka,
Vladislav Mladenov, Jörg Schwenk

hgi NDS

Real-Time Web Editing Tools



Horst Görtz Institut für IT-Sicherheit

Operational Transforms (OT)

- Maintain a consistent view on a document
- Automatically resolve editing conflicts
- Whole area of research on its own

Motivation

- Established tools do not apply (cryptographic) protection to documents
- Previous academic proposals with encryption either
  - Require large overheads
  - Are not real-time collaborative,
  - Require browser extensions, or
  - Do not take structure into account
- Is it feasible to have all these properties?

SECRET

- First **S**ecure, **E**fficient, and **C**ollaborative **R**eal-time **E**di**t**or
- SECRET is the first collaboration tool with
  1. encryption of whole documents or arbitrary sub-parts,
  2. novel combination of tree-based OT and structure preserving encryption,
  3. only a modern browser without any extra software installation or browser extension required

Building Blocks

- ShareJS
  - JavaScript Middleware with OT algorithms
- State-of-the-Art Web Technologies
  - WebSockets for Asynchronous Messaging
  - W3C Web Cryptography API for AES-128 in Galois Counter Mode (GCM)
  - PostMessage API
- XML Encryption
  - Structure Preserving Encryption

XML Encryption

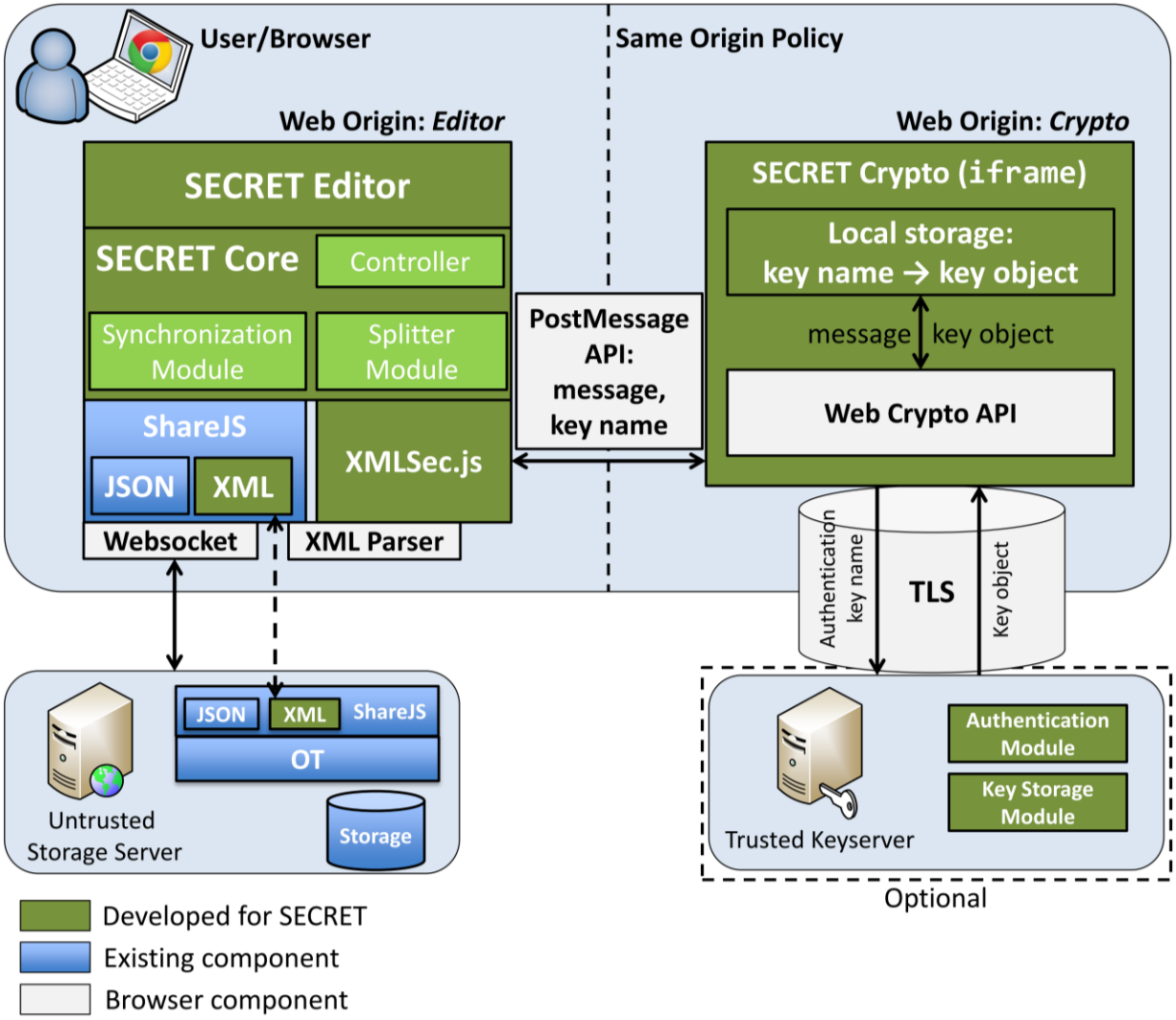
```
<PaymentInfo>
  <Name>John Smith</Name>
  <CreditCard Limit="5,000" Currency="USD">
    <Number>1234 5678 2580 1595</Number>
    <Issuer>Example Bank</Issuer>
    <Expiration>04/17</Expiration>
  </CreditCard>
</PaymentInfo>
```

```
<PaymentInfo>
  <Name>John Smith</Name>
  <EncryptedData
    Type="http://www.w3.org/2001/04/xmlenc#Element"
    xmlns="http://www.w3.org/2001/04/xmlenc#">
    <CipherData>
      <CipherValue>184797A8C2FE977DEFA10A7FE540A0D0</CipherValue>
    </CipherData>
  </EncryptedData>
</PaymentInfo>
```

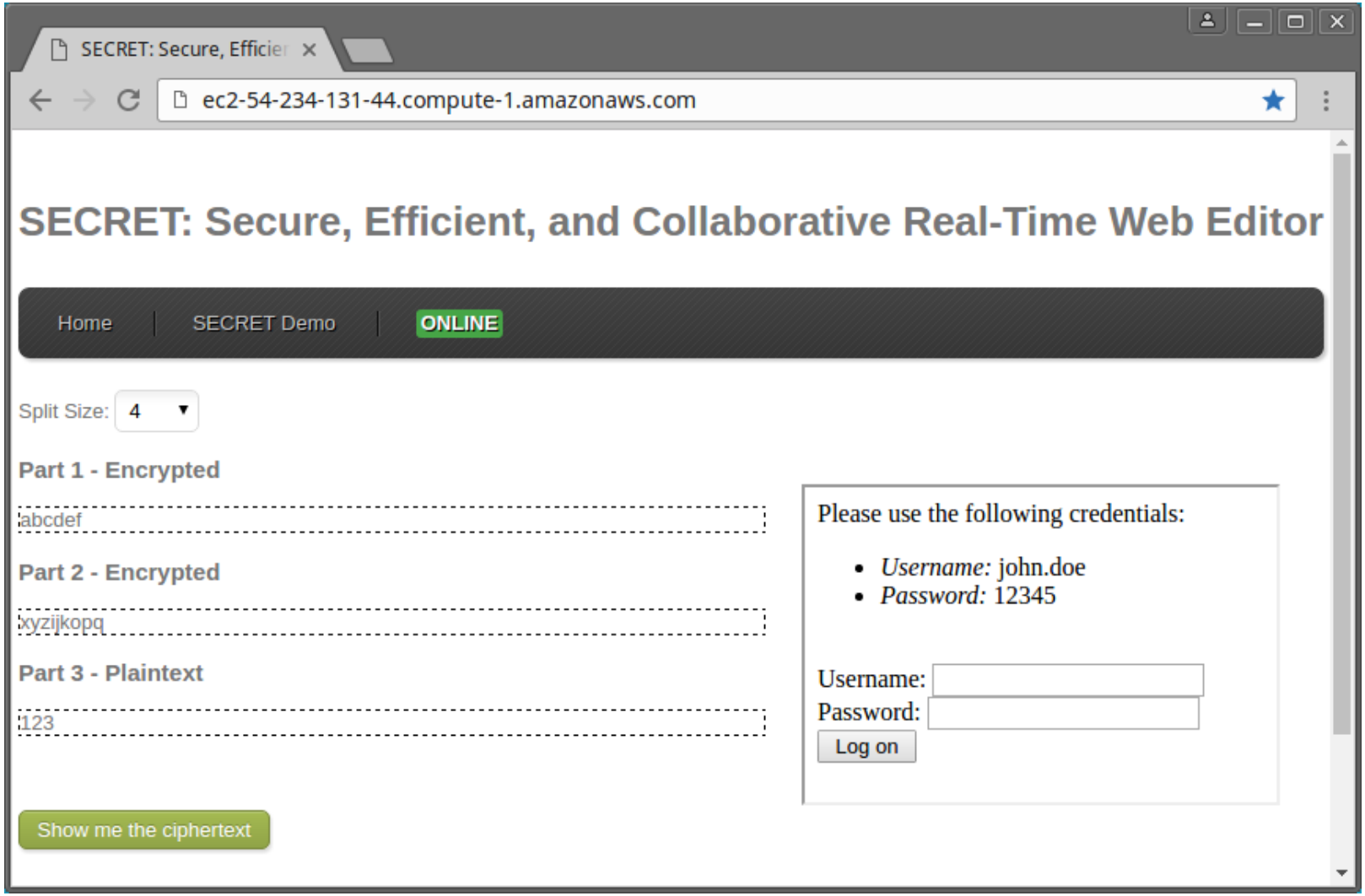
Implementation Challenges

- ShareJS does not support XML
  - Solution: Implemented OT for XML documents as an extension of ShareJS
- Browsers do not support XML Encryption
  - Solution: Implemented a JavaScript library to encrypt, decrypt, sign, or verify documents
- WebCrypto API does not handle long-lived, persistent keys
  - Solution: Store them on an key-server or derive them from a password

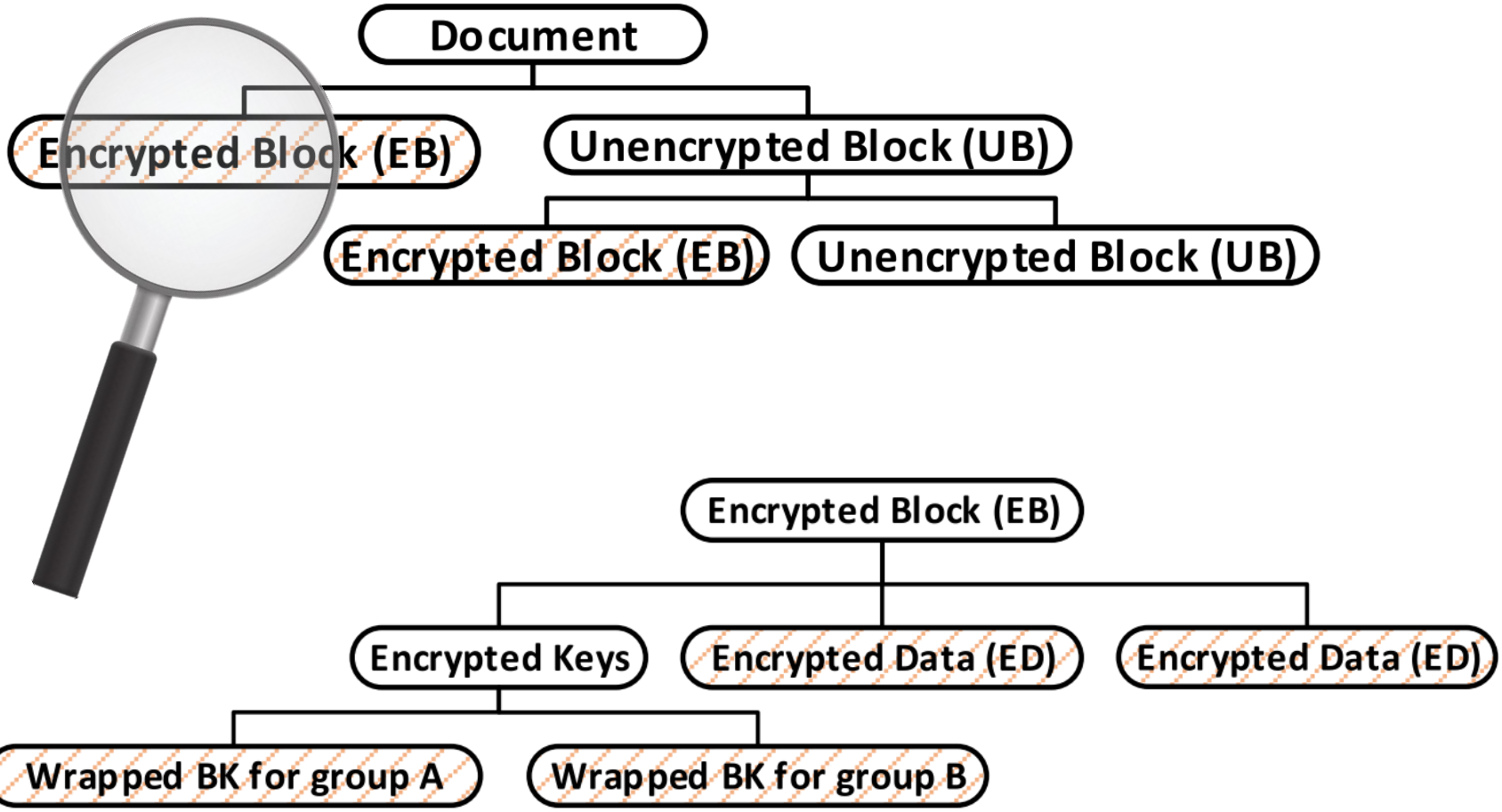
Architecture Overview



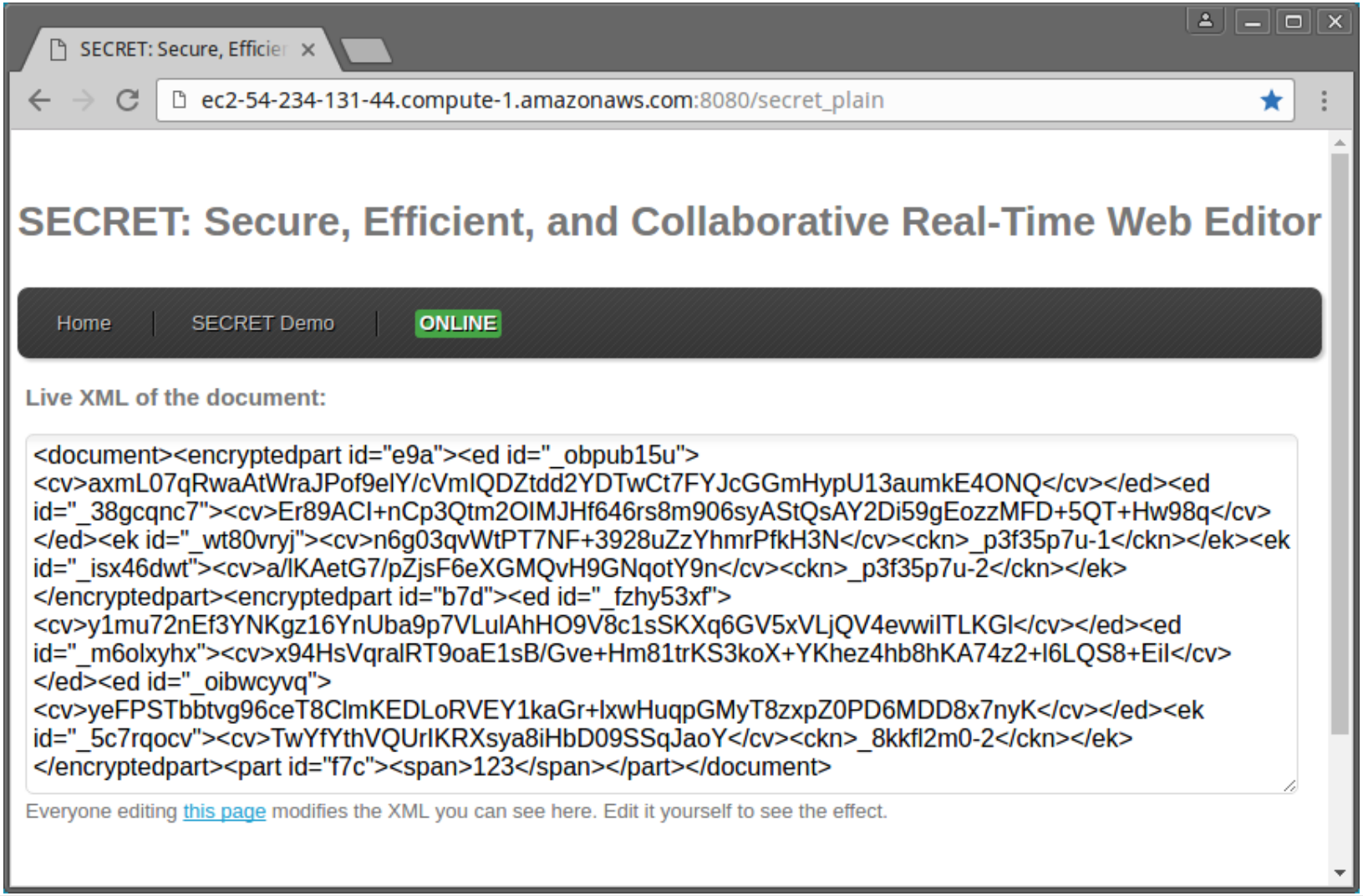
SECRET



Documents


Document
Encrypted Block (EB)
Unencrypted Block (UB)
Encrypted Block (EB)
Unencrypted Block (UB)
Encrypted Block (EB)
Encrypted Keys
Encrypted Data (ED)
Encrypted Data (ED)
Wrapped BK for group A
Wrapped BK for group B

Screenshot Ciphertext



Splitting into Encrypted Data Chunks

- Large encrypted blocks
  - ⇒ Updates are inefficient
  - ⇒ Splitting is necessary
- Small encrypted blocks
  - ⇒ Large XML overhead

```
<div>
  <span>Hello</span>
  <span>World</span>
</div>
```

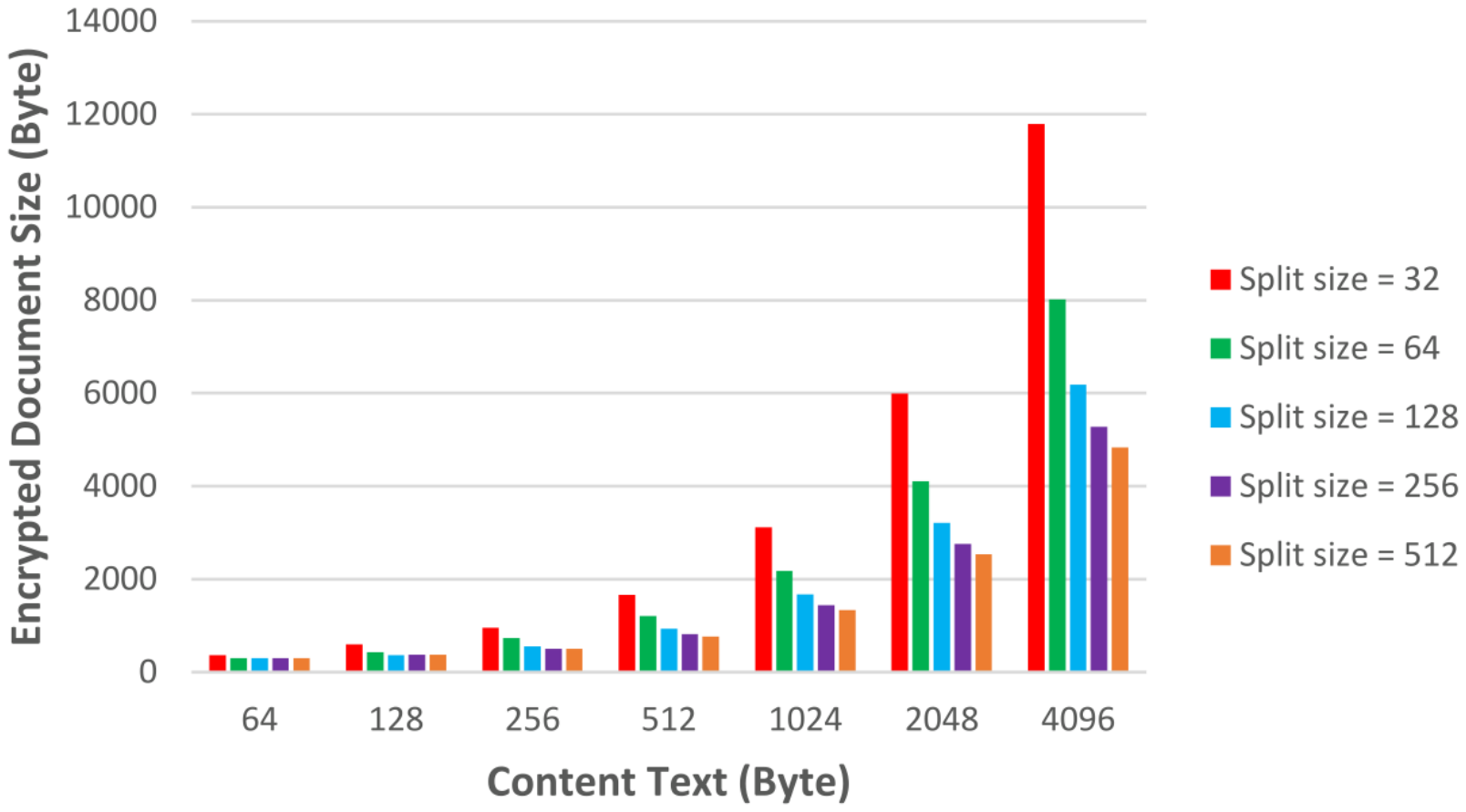
- Q: What is the optimal split size?

Horst Görtz Institut für IT-Sicherheit

Evaluation

- Google Chrome 50 with Selenium
- Simulated typing at 200 key strokes / min
- Measured storage and network overhead

Evaluation Storage



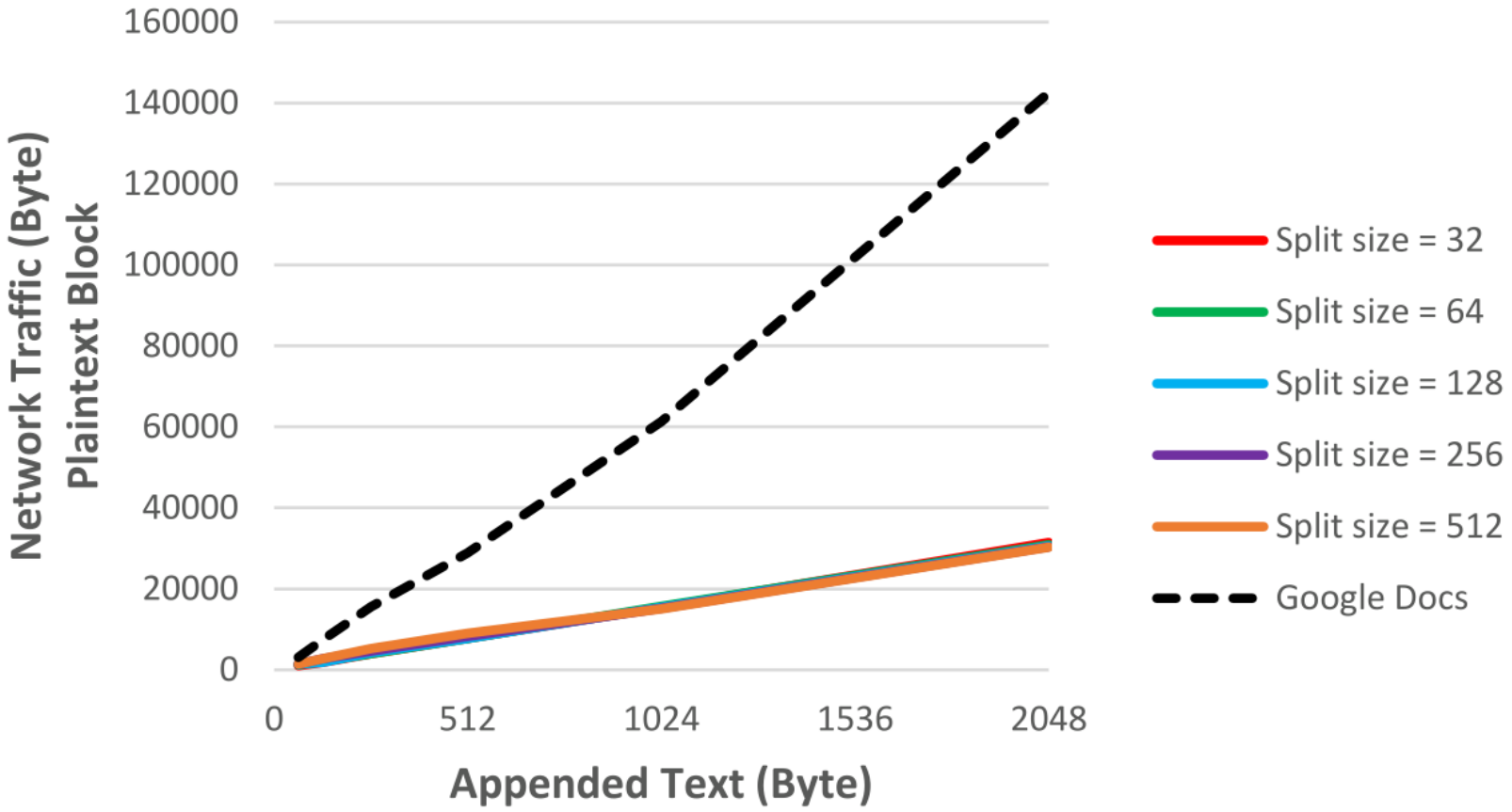
Evaluation Storage

Split size	32	64	128	256	512
Storage expansion	3.50	2.46	1.92	1.66	1.53

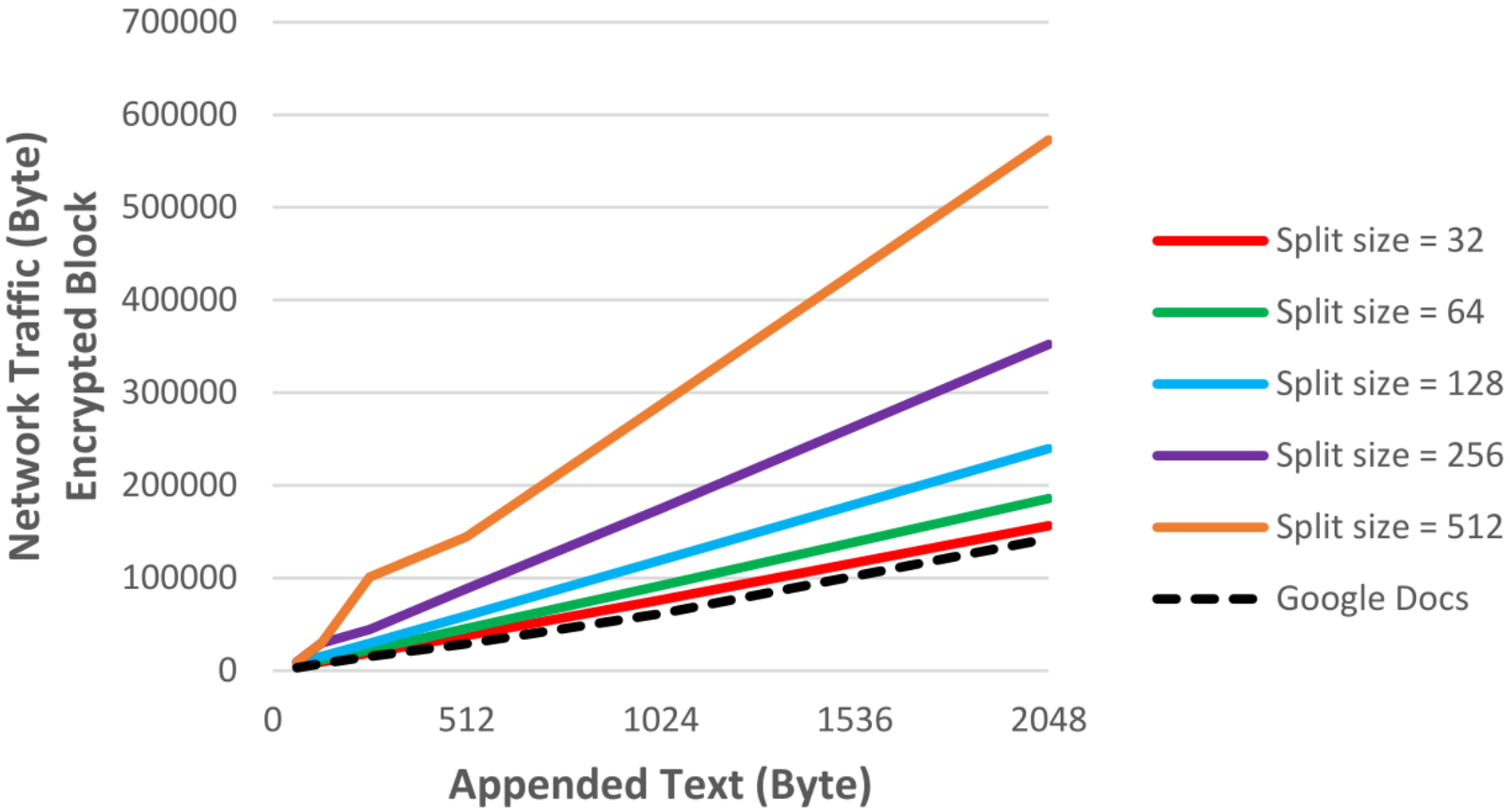
Table 2: Ciphertext expansion of a 4096 byte document.

- Numbers look high
- In fact, they are far below the numbers in related work
- Best results before ours: 3.75 – 4.82

Evaluation Network



Evaluation Network



Bandwidth Requirements

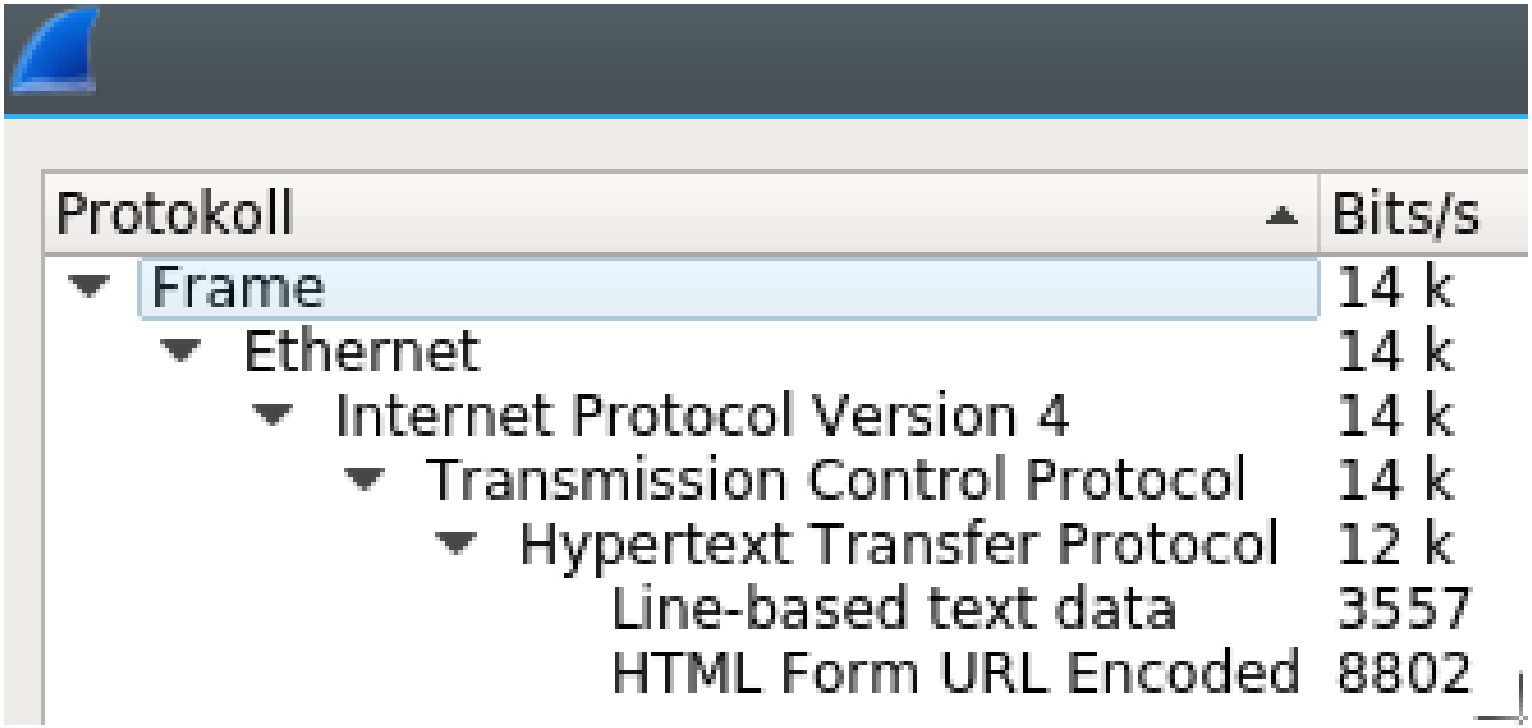


Figure 7: Screenshot from *Wireshark* measuring the required bandwidth for SECRET at 200 key strokes per minute with a split size of 128 bytes.

Conclusion & Outlook

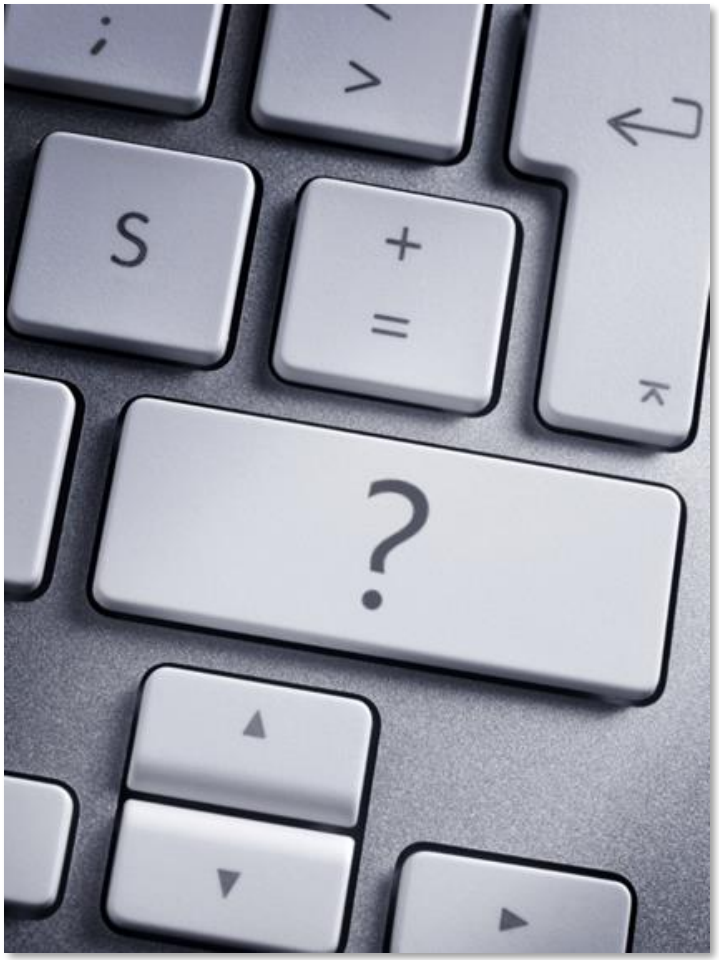
RUB

- A Secure, Efficient, and Collaborative Real-Time Web Editor is feasible
- No need for large overheads when using Structure Preserving Encryption

- GUI and editing features can be improved
- How about full-fledged office documents?
- SECRET's code is on GitHub:

  https://github.com/RUB-NDS/SECRET/

Questions?



Dennis Felsch
Horst Görtz Institute for IT-Security
Chair for Network and Data Security
Ruhr-University Bochum

dennis.felsch@rub.de

Twitter: @dfelsch

NDS-Blog:
http://web-in-security.blogspot.de