Software **SOLVED**

Developed with you

USING THE WEB CRYPTOGRAPHY API IN PWAS/SPAS

JON STACE

@JONSTACE

Introduction

- Increased use of PWAs and SPAs
- A need to encrypt data locally
- Options to achieve this
- Web browser based, not NodeJS

Why Not?

- 'JavaScript Cryptography Considered Harmful' https://www.nccgroup.trust/us/about-us/newsroom-and-events/blog/2011/august/javascript-cryptography-considered-harmful/
- Secure delivery of JavaScript to browsers
- Browser JavaScript is hostile to cryptography
  - Lack of secure random number generator
  - Malleability of JavaScript runtime
  - Lack of secure erase
  - Functions with known timing characteristics
  - A secure keystore

Pure JavaScript options

- There's a great big list:

https://gist.github.com/jo/8619441

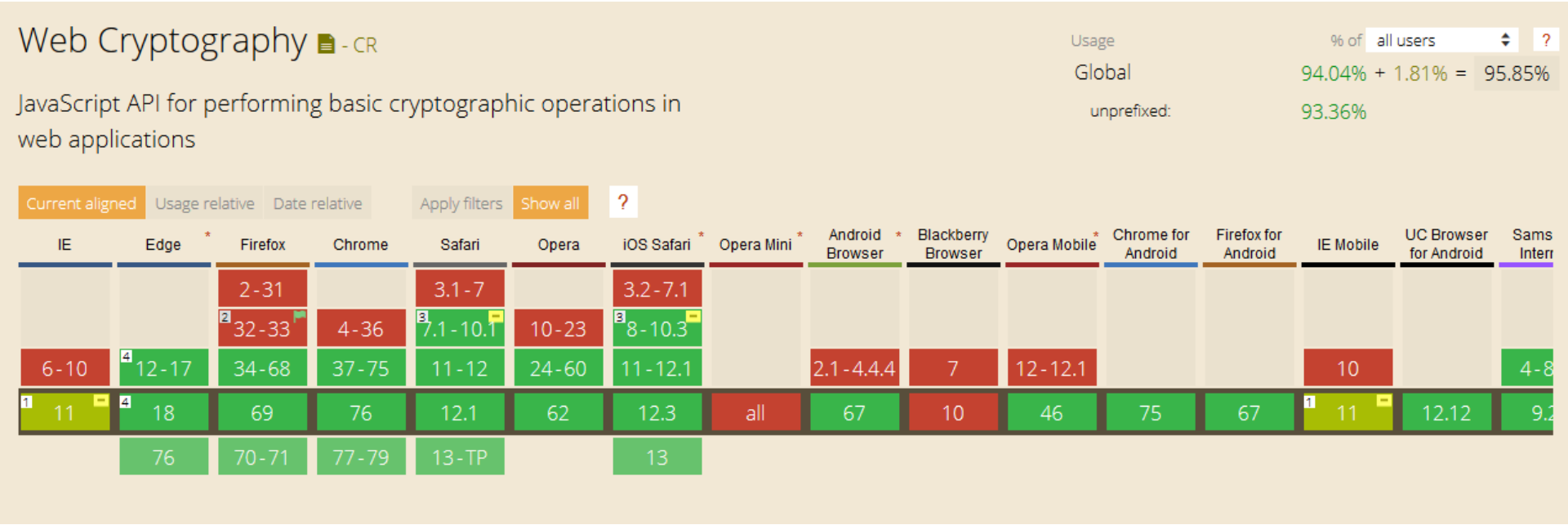
- Stanford Javascript Crypto Lib - http://bitwiseshiftleft.github.io/sjcl/
- Jsencrypt - https://github.com/travist/jsencrypt
- Libsodium - https://download.libsodium.org/doc/bindings_for_other_languages

- Don't create your own crypto!

WebCryptoAPI

- http://www.w3.org/TR/WebCryptoAPI/
- https://developer.mozilla.org/en-US/docs/Web/API/Web_Crypto_API
- Features
  - Random Number Generator
  - Secure key store
  - Multiple algorithms and modes

Browser Compatibility




You know nothing, Jon Snow.
Stace

Problems

- ‘subtle’ API
- Reliant on browser’s security of implementation, PRNG, cross-tab
- Fortunately, you can’t shim the crypto interface
- Can inspect browser’s memory and JavaScript runtime for unencrypted data
- Old browsers!

Test your browser!

- https://github.com/diafygi/webcrypto-examples

Code Demo

https://github.com/jonstace/WebCryptoAPIDemo

Summary

- WebCryptoAPI
- Modern browsers supported
- ‘Subtle’ API
- Secure PRNG and key store
- Hashing, signing and encryption algorithms
- Better than no encryption at rest – layered security

Questions?

https://www.slideshare.net/JonStace/jon-stace-web-cryptography-api-170657271