



Software SOLVED

Developed with you

USING THE WEB CRYPTOGRAPHY API IN PWAS/SPAS

JON STACE

@JONSTACE



Software SOLVED

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

Web Cryptography 📄 - CR

JavaScript API for performing basic cryptographic operations in web applications

Usage % of all users ?
Global 94.04% + 1.81% = 95.85%
unprefixed: 93.36%

Current aligned	Usage relative	Date relative	Apply filters	Show all	?										
IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Blackberry Browser	Opera Mobile *	Chrome for Android	Firefox for Android	IE Mobile	UC Browser for Android	Sams Intern
		2-31		3.1-7		3.2-7.1									
		32-33	4-36	7.1-10.1	10-23	8-10.3									
6-10	12-17	34-68	37-75	11-12	24-60	11-12.1		2.1-4.4.4	7	12-12.1			10		4-8
11	18	69	76	12.1	62	12.3	all	67	10	46	75	67	11	12.12	9.2
	76	70-71	77-79	13-TP		13									





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>



Software **SOLVED**