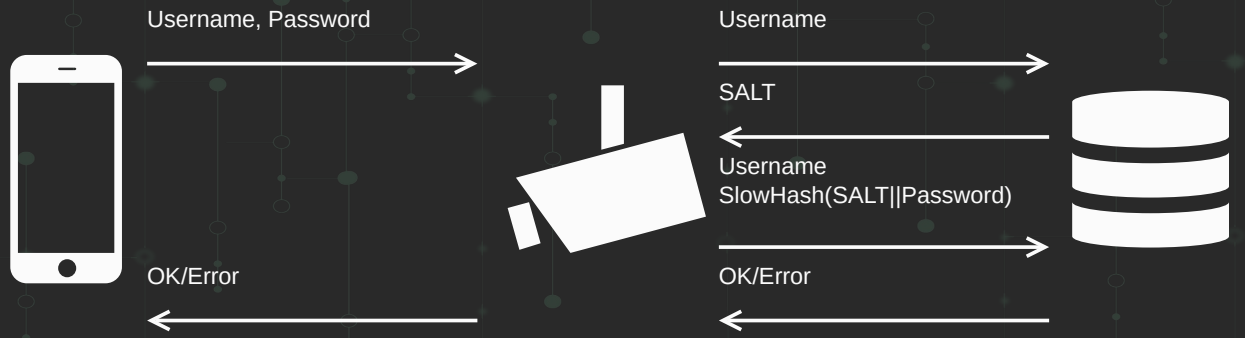
Clipaha: A Scheme to Perform Password Stretching on the Client

Francisco Blas Izquierdo Riera (Chalmers)
Magnus Almgren (Chalmers)
Pablo Picazo-Sanchez (Halmstad)
Christian Rohner (Uppsala)
Sponsored by MSB through the RIOT project

Password Authentication

- Passwords widely used for authentication
- Easy to implement
- (Almost) always with the user



Password Issues

- Reuse (Das et al., 2014)
- Low entropy (Weber et al., 2008)
- Leaks (Bauman, 2015)
- Cracking accelerators (Hatzivasilis 2017)

State of the Art

- Password Hashing Competition → Argon2 (Aumasson et al., 2013; Biryukov et al., 2017)
  - Recommends high memory bandwidth, space and CPU
- Argon2 hard to integrate (Wijayarathna et al., 2018)
- Server-side hashing most common (Blanchard et al., 2019)
- Problematic non-academic suggestions: Crackstation and Libsodium

Client-side password hashing still unsolved

Client-side password hashing is a set of known or at least folklore approaches.

— Reviewer #2

Threat model

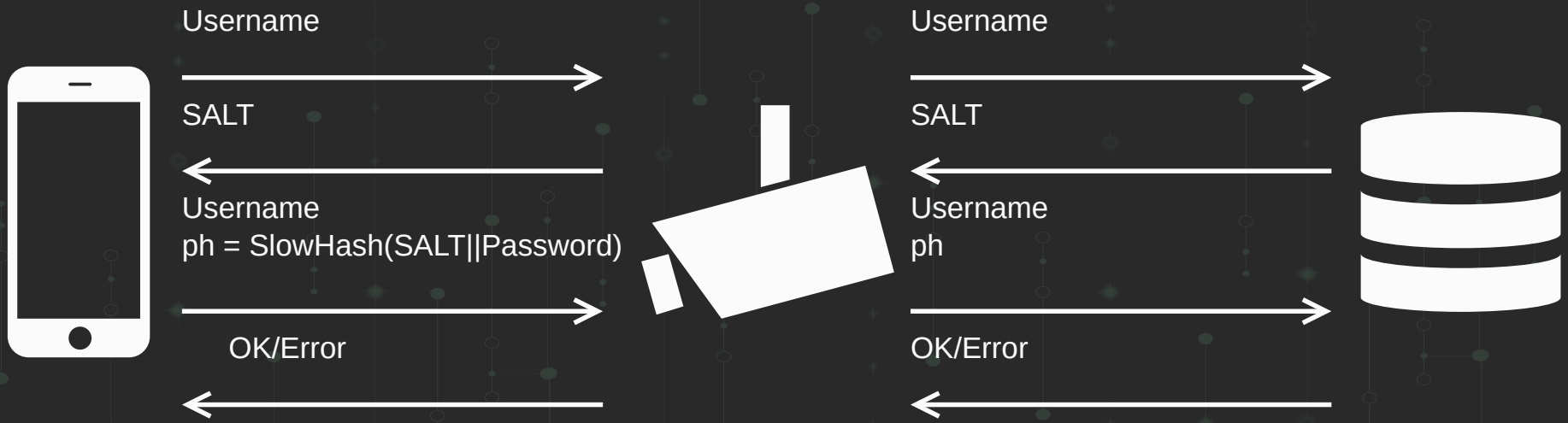
Online attacker

- Uses login
- Credential bruteforcing
  - User enumeration

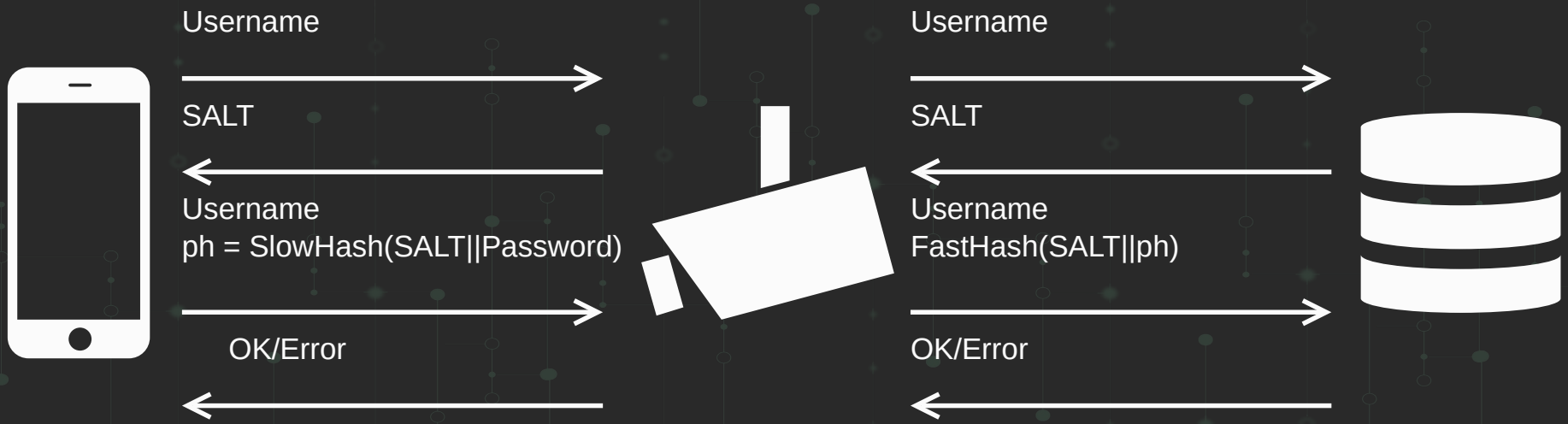
Offline attacker

- Leaked database
- Credential cracking
  - Salt collisions
- Pass-the-hash

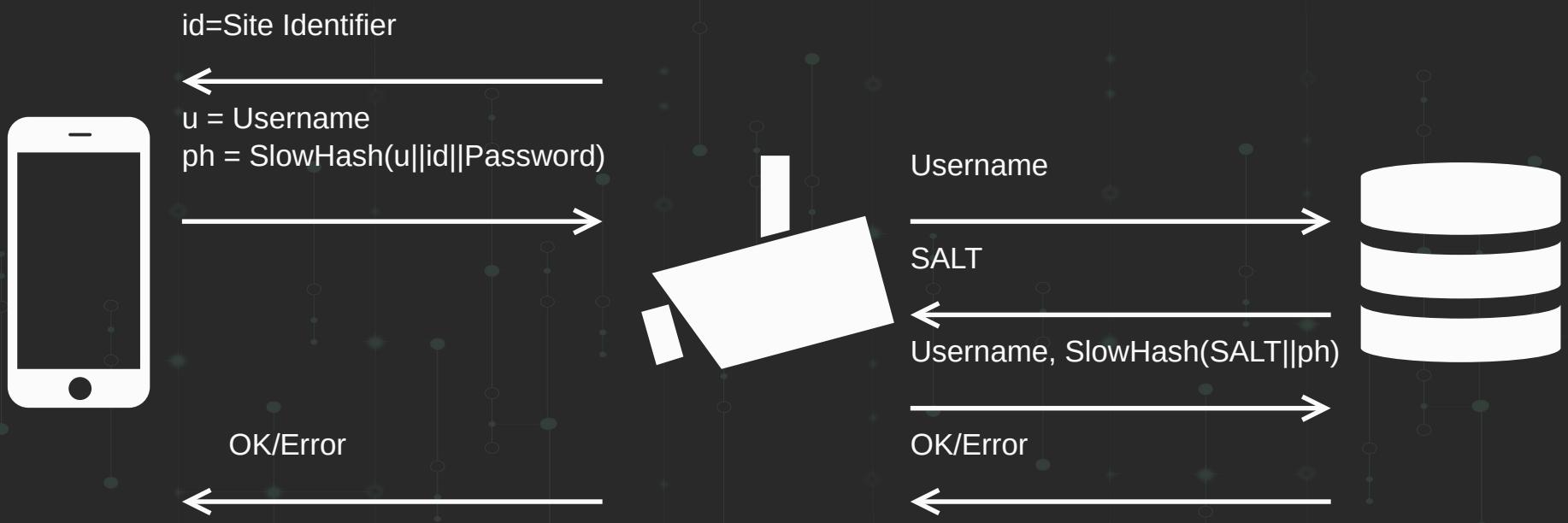
Trivial approach



Libsodium


Username
SALT
Username
ph = SlowHash(SALT||Password)
OK/Error
Username
SALT
Username
FastHash(SALT||ph)
OK/Error

Crackstation



Clipaha (our proposal)

Recommendations vs Clipaha

Implementation	Salt collisions	User Enumeration	Missing Reference Library
Crackstation	✗	✓	✗
Libsodium	✓	✗	✓*
Clipaha	✓	✓	✓

Clipaha's design requirements

- **RQ-S1** Resistance to salt collisions
- **RQ-S2** Resistance to user enumeration
- **RQ-S3** Resistance to pass-the-hash attacks
- **RQ-S4** Ensure the algorithm is at least as secure as if performed by the server
- **RQ-F1** Not require software developers to choose security relevant parameters
- **RQ-F2** Suitable for the many contexts where passwords are used
- **RQ-P1** Limit the delay of the authentication process for legitimate users

Security levels

Level	low	medium	high	ultra
Iterations	6	5	3	3
Memory (MiB)	192	384	1024	2016

Security levels of Clipaha

Web as IoT prototype

- Web is a common way to administer IoT
- Allows client-side code execution
- Allows reuse for other platforms

Issues Running Password Hashing on Browsers

- Memory limitations (mobile 1GiB, desktop 2GiB)
- Memory fragmentation (32-bit)
- Multithreading
- Language performance (JavaScript)
- Data remanence

Native vs Wasm vs JS

		low		medium		high		ultra	
Brow.	Impl.	Avg (s)	Factor	Avg (s)	Factor	Avg (s)	Factor	Avg (s)	Factor
Nat	AVX2	0.653	1	1.446	1	2.083	1	4.068	1
Chro	WA	1.873	2.87	3.181	2.20	5.269	2.53	10.845	2.67
	JS	80.507	123.29	133.685	92.45	215.930	103.66	422.795	103.93
FF	WA	2.228	3.41	3.770	2.61	6.040	2.90	11.912	2.93
	JS	29.109	44.58	47.909	33.13	77.254	37.09	151.300	37.19

Core i7-4710HQ with 16 GiB of RAM Chromium 85.0.4183.83 / Firefox 68.12.0 ESR

CHALMERS
UNIVERSITY OF TECHNOLOGY

Garmer technologies

Chrome has slow asm.js

		low		medium		high		ultra	
Brow.	Impl.	Avg (s)	Factor	Avg (s)	Factor	Avg (s)	Factor	Avg (s)	Factor
Nat	AVX2	0.653	1	1.446	1	2.083	1	4.068	1
Chro	WA	1.873	2.87	3.181	2.20	5.269	2.53	10.845	2.67
	JS	80.507	**123.29**	133.685	**92.45**	215.930	**103.66**	422.795	**103.93**
FF	WA	2.228	3.41	3.770	2.61	6.040	2.90	11.912	2.93
	JS	29.109	44.58	47.909	33.13	77.254	37.09	151.300	37.19

Core i7-4710HQ with 16 GiB of RAM Chromium 85.0.4183.83 / Firefox 68.12.0 ESR

Webassembly up to 3.41x slower

		low		medium		high		ultra	
Brow.	Impl.	Avg (s)	Factor	Avg (s)	Factor	Avg (s)	Factor	Avg (s)	Factor
Nat	AVX2	0.653	1	1.446	1	2.083	1	4.068	1
Chro	WA	1.873	2.87	3.181	2.20	5.269	2.53	10.845	2.67
	JS	80.507	123.29	133.685	92.45	215.930	103.66	422.795	103.93
FF	WA	2.228	**3.41**	3.770	2.61	6.040	2.90	11.912	2.93
	JS	29.109	44.58	47.909	33.13	77.254	37.09	151.300	37.19

Core i7-4710HQ with 16 GiB of RAM Chromium 85.0.4183.83 / Firefox 68.12.0 ESR

CHALMERS UNIVERSITY OF TECHNOLOGY

Garmer technologies

Chrome has faster WA

		low		medium		high		ultra	
Brow.	Impl.	Avg (s)	Factor	Avg (s)	Factor	Avg (s)	Factor	Avg (s)	Factor
Nat	AVX2	0.653	1	1.446	1	2.083	1	4.068	1
Chro	WA	**1.873**	**2.87**	**3.181**	**2.20**	**5.269**	**2.53**	**10.845**	**2.67**
	JS	80.507	123.29	133.685	92.45	215.930	103.66	422.795	103.93
FF	WA	**2.228**	**3.41**	**3.770**	**2.61**	**6.040**	**2.90**	**11.912**	**2.93**
	JS	29.109	44.58	47.909	33.13	77.254	37.09	151.300	37.19

Core i7-4710HQ with 16 GiB of RAM Chromium 85.0.4183.83 / Firefox 68.12.0 ESR

Failure rates

Device	Library	Dev. count	low	medium	high	ultra
Phone	clipaha	15	0	1	13	13
	libsodium		0	2	13	13
Tablet	clipaha	2	0	0	0	1
	libsodium		0	0	1	1
Computer	clipaha	18	0	0	0	0
	libsodium		0	0	0	0

Number of failures per device type, security level and library

No failures on low

Device	Library	Dev. count	low	medium	high	ultra
Phone	clipaha	15	**0**	1	13	13
	libsodium		**0**	2	13	13
Tablet	clipaha	2	**0**	0	0	1
	libsodium		**0**	0	1	1
Computer	clipaha	18	**0**	0	0	0
	libsodium		**0**	0	0	0

Number of failures per device type, security level and library

Clipaha fails less

Device	Library	Dev. count	low	medium	high	ultra
Phone	clipaha	15	0	**1**	13	13
	libsodium		0	**2**	13	13
Tablet	clipaha	2	0	0	**0**	1
	libsodium		0	0	**1**	1
Computer	clipaha	18	0	0	0	0
	libsodium		0	0	0	0

Number of failures per device type, security level and library

High and ultra fail similarly

Device	Library	Dev. count	low	medium	high	ultra
Phone	clipaha	15	0	1	**13**	**13**
	libsodium		0	2	**13**	**13**
Tablet	clipaha	2	0	0	**0**	**1**
	libsodium		0	0	**1**	**1**
Computer	clipaha	18	0	0	0	0
	libsodium		0	0	0	0

Number of failures per device type, security level and library

Benchmark results

Device	Library	low		medium		high		ultra	
		Max	Min	Max	Min	Max	Min	Max	Min
Phone	clipaha	19.134	7.362	31.571	12.228	25.900	20.012	54.175	41.648
	libsodium	33.313	12.611	54.887	21.076	48.320	34.648	100.377	69.114
Tablet	clipaha	2.279	2.015	4.048	3.739	7.689	5.940	17.674	17.674
	libsodium	4.287	3.440	7.253	5.739	11.913	11.913	25.857	25.857
Computer	clipaha	3.032	1.1126	4.979	1.873	3.328	3.328	15.717	6.124
	libsodium	6.654	1.993	11.097	3.348	5.464	5.464	34.015	11.080

Time (s) per device type, security level and library. Average of five runs.

Low < 20s

Device	Library	low		medium		high		ultra	
		Max	Min	Max	Min	Max	Min	Max	Min
Phone	clipaha	**19.134**	7.362	31.571	12.228	25.900	20.012	54.175	41.648
	libsodium	33.313	12.611	54.887	21.076	48.320	34.648	100.377	69.114
Tablet	clipaha	**2.279**	2.015	4.048	3.739	7.689	5.940	17.674	17.674
	libsodium	4.287	3.440	7.253	5.739	11.913	11.913	25.857	25.857
Computer	clipaha	**3.032**	1.1126	4.979	1.873	3.328	3.328	15.717	6.124
	libsodium	6.654	1.993	11.097	3.348	5.464	5.464	34.015	11.080

Time (s) per device type, security level and library. Average of five runs.

Clipaha faster than libsodium

Device	Library	low		medium		high		ultra	
		Max	Min	Max	Min	Max	Min	Max	Min
Phone	clipaha	19.134	7.362	31.571	12.228	25.900	20.012	54.175	41.648
	libsodium	33.313	12.611	54.887	21.076	48.320	34.648	100.377	69.114
Tablet	clipaha	2.279	2.015	4.048	3.739	7.689	5.940	17.674	17.674
	libsodium	4.287	3.440	7.253	5.739	11.913	11.913	25.857	25.857
Computer	clipaha	3.032	1.1126	4.979	1.873	3.328	3.328	15.717	6.124
	libsodium	6.654	1.993	11.097	3.348	5.464	5.464	34.015	11.080

Time (s) per device type, security level and library. Average of five runs.

Phones significantly slower

Device	Library	low		medium		high		ultra	
		Max	Min	Max	Min	Max	Min	Max	Min
Phone	clipaha	**19.134**	**7.362**	**31.571**	**12.228**	**25.900**	**20.012**	**54.175**	**41.648**
	libsodium	33.313	12.611	54.887	21.076	48.320	34.648	100.377	69.114
Tablet	clipaha	2.279	2.015	4.048	3.739	7.689	5.940	17.674	17.674
	libsodium	4.287	3.440	7.253	5.739	11.913	11.913	25.857	25.857
Computer	clipaha	3.032	1.1126	4.979	1.873	3.328	3.328	15.717	6.124
	libsodium	6.654	1.993	11.097	3.348	5.464	5.464	34.015	11.080

Time (s) per device type, security level and library. Average of five runs.

Real Scenarios

- Registration
- Login
- Password change
- Client-side caching
- Password managers
- Non-human authentication
- PAKE
- MFA

Registration

- Send also plaintext password for policy enforcement
- Prevent user enumeration by not disclosing if user already registered
- Timing must be the same

Login

- Client send hashed password
- Server checks in database
- Failures must not distinguish username presence
- Timing must be the same

Client-side caching

- Useful to reduce CPU usage
- Reducing response time may open side-channel
- Cached credentials protected same as passwords

Password managers

- Client code can store and retrieve hashes
- Hash storage must be as secure as password
- Policy for hash access must be same as for password

Non-human authentication (IoT client)

- Use high entropy token instead of hash
- Token must be protected with same security as password

Password Authenticated Key Exchanges (PAKE)

- Clipaha can be used as preprocessing step for password
- Timing relevant to avoid side-channels
- PAKE needs to ensure server verifiers cannot be reused

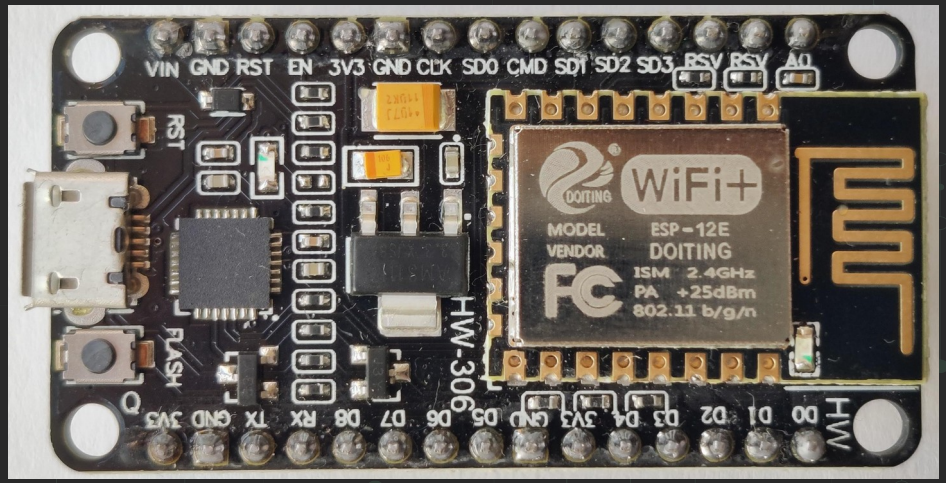
Multi Factor Authentication (MFA)

- Codes provided with password
- If action needed perform after verification
- Ensure constant time operation

Conclusions

- Clipaha provides readily available client-side password hashing for IoT servers
- Clipaha resists user enumeration and salt collisions
- Clipaha can run on most devices from the last 5 years
- Clipaha provides new, secure scenarios leveraging client-side password hashing

Demo



http://demo.clipaha.com:1487

CHALMERS
UNIVERSITY OF TECHNOLOGY

Garmer technologies

CHALMERS
UNIVERSITY OF TECHNOLOGY

technologies
Garmer

QUESTIONS?