EXPLORING PASSWORD-AUTHENTICATED KEY-EXCHANGE ALGORITHMS

A project report submitted to The University of Manchester
for the degree of Bachelors of Science
in the Faculty of Science and Engineering

2023

Sam Leonard
f41751sl
Supervisor: Professor Bernardo Magri

Department of Computer Science

Declaration

No portion of the work referred to in this project report has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning.

Copyright

Abstract

Exploring Password-Authenticated Key-Exchange Algorithms

Sam Leonard, Supervisor: Professor Bernardo Magri

Password-Authenticated Key-Exchange (PAKE) algorithms are a niche kind of cryptography where parties seek to establish a strong shared key, from a low entropy secret such as a password. This makes it particularly attractive to some domains, such as Industrial Internet of Things (IIOT). However many PAKE algorithms are unsuitable for Internet of Things (IOT) applications, due to their heavy computational requirements. Augmented Composable Password Authenticated Connection Establishment (AuCPace) is a new PAKE protocol which aims to make PAKEs accessible to IIOT by utilising Elliptic Curve Cryptography (ECC), Verifier based PAKEs (V-PAKEs) and a novel augmented approach. This project aims to provide an approachable and developer-focused implementation of AuCPace in Rust and to contribute this implementation back to RustCrypto to promote wider adoption of PAKE algorithms.

Acknowledgements

I would like to thank everyone at absolutely wonderful supervisor, RustCrypto, Crypto Hack and my CTF Team (0rganizers). Without your help none of this would have been possible.

Contents

CHAPTER 2

Design

2.1 Why Rust?

AuCPace explicitly targets IIOT in it's design. Rust is rapidly becoming a popular choice for IOT and embedded software applications. This is due to it's focus on memory safety, developer experience and it's strong embedded ecosystem. Libraries like Embassy and RTIC allow the user to program high level logic and use powerful abstractions to interact with the hardware through Rust objects, while still compiling down to small efficient binaries. Embassy is especially impressive as they have implemented a async executor so that multitasking in embedded applications can be performed with the same async/await framework that programmers are familiar with. A short Embassy examples is shown in listing 1. Tools such as `probe-rs` allow developers to maintain the same workflow they would when working on a normal rust binary, by implementing a `cargo` runner which flashes the binary to the embedded device then using Real-Time-Transfer (RTT) to receive debug messages from the device. Those debug messages can be setup automatically using libraries such as `defmt_rtt` which use RTT to send a compressed representation of the debug message to be formatted later on using a technique called deferred formatting, allowing for debug messages to take up a fraction of the size of the original message. Together

Listing 1: Embassy async/await example

```rust
use defmt::info;
use embassy::executor::Spawner;
use embassy::time::{Duration, Timer};
use embassy_nrf::gpio::{AnyPin, Input, Level, Output, OutputDrive, Pin,
↪ Pull};
use embassy_nrf::Peripherals;

// Declare async tasks
#[embassy::task]
async fn blink(pin: AnyPin) {
    let mut led = Output::new(pin, Level::Low, OutputDrive::Standard);

    loop {
```

```
        // Timekeeping is globally available, no need to mess with hardware
 ↪  timers.
        led.set_high();
        Timer::after(Duration::from_millis(150)).await;
        led.set_low();
        Timer::after(Duration::from_millis(150)).await;
    }
}

// Main is itself an async task as well.
#[embassy::main]
async fn main(spawner: Spawner, p: Peripherals) {
    // Spawned tasks run in the background, concurrently.
    spawner.spawn(blink(p.P0_13.degrade())).unwrap();

    let mut button = Input::new(p.P0_11, Pull::Up);
    loop {
        // Asynchronously wait for GPIO events, allowing other tasks
        // to run, or the core to sleep.
        button.wait_for_low().await;
        info!("Button pressed!");
        button.wait_for_high().await;
        info!("Button released!");
    }
}
```

2.2 Developer Focussed Design

Glossary

Abelian Group A group whose operator is also commutative. e.g. Addition over $\mathbb{Z}$. . 12, 13

AES Advanced Encryption Scheme. 31

AKE Authenticated Key-Exchange. 15

Asymmetric Cryptography Asymmetric Cryptography is where the the sender and receiver each have two keys - a public key which can be freely shared, and a private key which must be kept secret. Common examples of this are the RSA scheme and the various DH flavours. 9, 10

AuCPace Augmented Composable Password Authenticated Connection Establishment. 4, 15, 16, 17, 18, 21, 24, 41

Augmented PAKE A Balanced PAKE is one in which both parties share knowledge the same secret. This is in contrast to other schemes such as Verifier-based/Augmented PAKEs. . 8, 11, 14, 15, 30

Balanced PAKE A Balanced PAKE is one in which both parties share knowledge the same secret. This is in contrast to other schemes such as Verifier-based/Augmented PAKEs. . 8, 10, 14, 15

CFRG Crypto Forum Research Group. 14, 30

CPace Composable Password Authenticated Connection Establishment. 14, 17

DH Diffie-Hellman. 10, 13, 15, 16, 38

EAP Extensible Authentication Protocol. 9

ECC Elliptic Curve Cryptography. 4, 14

ECDLP Elliptic Curve Discrete Logarithm Problem. 14

EKE Encrypted Key Exchange. 9, 10, 11

FFI Foreign Function Interface. 22

Finite Field A Finite Field is a finite set with an associated addition and multiplication operator, where the operators satisfy the field axioms. Namely they are: Associative, Commutative, Distributive, they have inverses and identity elements. 13

HMI human machine interface. 15

IETF Internet Engineering Task Force. 14, 15

IIOT Industrial Internet of Things. 4, 15, 16, 24

IOT Internet of Things. 4, 24

iPAKE identity-binding PAKE. 15

IRTF Internet Research Task Force. 14

KHAPE Key-Hiding Asymmetric PAKE. 15, 16

MCU Microcontroller. 41

NIST National Institute of Standards and Technology. 12

nonce number used only once – A cryptographic term which relates to an ephemeral secret value, an example would be an Initialisation Vector for AES-CBC mode encryption. . 16

Online Cryptography Online cryptography is where interactions with the cryptosystem are only possible via real-time interactions with the server. Primarily this is to prevent offline computation. 8, 10

OPAQUE An Asymmetric PAKE Protocol Secure Against Pre-Computation Attacks. Augmented PAKE Winner of the Crypto Forum Research Group (CFRG) PAKE selection process. The name is a play on words from OPAKE, where O is Oblivious Pseudo Random Function (OPRF). 14, 15, 16

OPRF Oblivious Pseudo Random Function. 15, 30

PAKE Password-Authenticated Key-Exchange. 4, 7, 8, 9, 11, 14, 15, 16, 22

PKI Public-Key-Infrastructure. 15, 16

PRS Password Related String. 17

PSK Pre-Shared Key. 16

RSA Rivest-Shamir-Adleman. 9, 12

RTT Real-Time-Transfer. 24

Safe Prime A number $2n + 1$ is a Safe Prime if n is prime, it is the effectively the other part of a Sophie Germain prime. . 10, 11

SPAKE Simple PAKE. 10, 11

SRP Secure Remote Password. 11, 38

SSID Sub-Session ID. 16, 17

Symmetric Cryptography Symmetric Cryptography is where the both the sender and receiver share the same secret key. It is normally computationally more efficient, the most common such scheme is Advanced Encryption Scheme (AES). 9

TLS Transport Layer Security. 7, 11

Verifier A representation of the user's password put through some one-way function. This could be as simple as just storing a hash of the password, though for most PAKEs the verifier is an element of whatever group we are working in. An example can be seen on page 11. 8, 11

V-PAKE Verifier based PAKE. 4

Bibliography

[AP05] Michel Abdalla and David Pointcheval. "Simple Password-Based Encrypted Key Exchange Protocols". In: *Topics in Cryptology – CT-RSA 2005*. Ed. by Alfred Menezes. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 191–208. ISBN: 978-3-540-30574-3.

[Ber+13] Daniel J Bernstein et al. "Elligator: elliptic-curve points indistinguishable from uniform random strings". In: *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*. 2013, pp. 967–980.

[BL] Daniel J. Bernstein and Tanja Lange. *SafeCurves: choosing safe curves for elliptic-curve cryptography.* Accessed 3rd April 2023. URL: https://safecurves.cr.yp.to.

[BL13] Daniel J Bernstein and Tanja Lange. "Security dangers of the NIST curves". In: *Invited talk, International State of the Art Cryptography Workshop, Athens, Greece*. 2013.

[BL17] Daniel J. Bernstein and Tanja Lange. *Montgomery curves and the Montgomery ladder*. Cryptology ePrint Archive, Paper 2017/293. 2017. URL: https://eprint.iacr.org/2017/293.

[BM92] Steven Michael Bellovin and Michael Merritt. *Encrypted key exchange: Password-based protocols secure against dictionary attacks*. May 1992.

[BMM00] Ingrid Biehl, Bernd Meyer, and Volker Müller. "Differential fault attacks on elliptic curve cryptosystems". In: *Advances in Cryptology—CRYPTO 2000: 20th Annual International Cryptology Conference Santa Barbara, California, USA, August 20–24, 2000 Proceedings 20*. Springer. 2000, pp. 131–146.

[Cha20] CFRG Chairs. *Results of the PAKE selection process*. Apr. 2020. URL: https://datatracker.ietf.org/meeting/interim-2020-cfrg-01/materials/slides-interim-2020-cfrg-01-sessa-results-of-the-pake-selection-process-00.

[Cre+20] Cas Cremers et al. *CHIP and CRISP: Protecting All Parties Against Compromise through Identity-Binding PAKEs*. Cryptology ePrint Archive, Paper 2020/529. 2020. URL: https://eprint.iacr.org/2020/529.

[GJK21] Yanqi Gu, Stanislaw Jarecki, and Hugo Krawczyk. *KHAPE: Asymmetric PAKE from Key-Hiding Key Exchange*. Cryptology ePrint Archive, Paper 2021/873. 2021. URL: https://eprint.iacr.org/2021/873.

[Gre18] Matthew Green. *Should you use SRP?* Accessed 3rd April, 2023. 2018. URL: https://blog.cryptographyengineering.com/should-you-use-srp/.

[Har08] Dan Harkins. "Simultaneous authentication of equals: A secure, password-based key exchange for mesh networks". In: *2008 Second International Conference on Sensor Technologies and Applications (sensorcomm 2008)*. IEEE. 2008, pp. 839–844.

[HL18] Björn Haase and Benoît Labrique. *AuCPace: Efficient verifier-based PAKE protocol tailored for the IIoT*. Cryptology ePrint Archive, Paper 2018/286. 2018. URL: https://eprint.iacr.org/2018/286.

[IT02] Tetsuya Izu and Tsuyoshi Takagi. "Exceptional procedure attack on elliptic curve cryptosystems". In: *Public Key Cryptography—PKC 2003: 6th International Workshop on Practice and Theory in Public Key Cryptography Miami, FL, USA, January 6–8, 2003 Proceedings 6*. Springer. 2002, pp. 224–239.

[Jea16] Jérémy Jean. *TikZ for Cryptographers*. https://www.iacr.org/authors/tikz/. 2016.

[KMV00] Neal Koblitz, Alfred Menezes, and Scott Vanstone. "The state of elliptic curve cryptography". In: *Designs, codes and cryptography* 19 (2000), pp. 173–193.

[Kob87] Neal Koblitz. "Elliptic curve cryptosystems". In: *Mathematics of computation* 48.177 (1987), pp. 203–209.

[Kra05] Hugo Krawczyk. "HMQV: A high-performance secure Diffie-Hellman protocol". In: *Crypto*. Vol. 3621. Springer. 2005, pp. 546–566.

[LL97] Chae Hoon Lim and Pil Joong Lee. "A key recovery attack on discrete log-based schemes using a prime order subgroup". In: *Advances in Cryptology–CRYPTO'97: 17th Annual International Cryptology Conference Santa Barbara, California, USA August 17–21, 1997 Proceedings 17*. Springer. 1997, pp. 249–263.

[Mil86] Victor S. Miller. "Use of Elliptic Curves in Cryptography". In: *Advances in Cryptology — CRYPTO '85 Proceedings*. Ed. by Hugh C. Williams. Berlin, Heidelberg: Springer Berlin Heidelberg, 1986, pp. 417–426. ISBN: 978-3-540-39799-1.

[MVO91] Alfred Menezes, Scott Vanstone, and Tatsuaki Okamoto. "Reducing elliptic curve logarithms to logarithms in a finite field". In: *Proceedings of the twenty-third annual ACM symposium on Theory of computing*. 1991, pp. 80–89.

[Pol78] John M Pollard. "Monte Carlo methods for index computation ($\mod p$)". In: *Mathematics of computation* 32.143 (1978), pp. 918–924.

[Sec21] Apple Platform Security. *Escrow security for iCloud Keychain*. Accessed 3rd April, 2023. May 2021. URL: https://support.apple.com/en-gb/guide/security/sec3e341e75d/web.

[Sem98] Igor Semaev. "Evaluation of discrete logarithms in a group of ρ-torsion points of an elliptic curve in characteristic ρ". In: *Mathematics of computation* 67.221 (1998), pp. 353–356.

[She+11] Y. Sheffer et al. *An EAP Authentication Method Based on the Encrypted Key Exchange (EKE) Protocol*. rfc 6124. RFC Editor, Feb. 2011. URL: https://www.rfc-editor.org/rfc/rfc6124.txt.

[ST20] National Institute of Standards and Technology. *Recommendation for Key Management: Part 1 – General*. Tech. rep. Federal Information Processing Standards Publications (FIPS PUBS) 140-2, Change Notice 2 December 03, 2002. Washington, D.C.: U.S. Department of Commerce, 2020. DOI: 10.6028/10.6028/NIST.SP.800-57pt1r5.

[Vol+04] John Vollbrecht et al. *Extensible Authentication Protocol (EAP)*. rfc 3748. RFC Editor, June 2004. URL: https://www.rfc-editor.org/rfc/rfc3748.txt.

[Wu+07] T. Wu et al. *Using the Secure Remote Password (SRP) Protocol for TLS Authentication*. RFC 5054. RFC Editor, Nov. 2007. URL: https://www.rfc-editor.org/rfc/rfc5054.txt.

[Wu00] Tom Wu. *The SRP Authentication and Key Exchange System*. RFC 2945. RFC Editor, Sept. 2000. URL: https://www.rfc-editor.org/rfc/rfc2945.txt.