Faculty of Electrical Engineering and Informatics
Technical University of Košice

Compact OPAQUE Protocol Implementation for Embedded Cryptographic Applications

Ing. Patrik Zeleňák   prof. Ing. Miloš Drutarovský, CSc.

Department of Electronics and Multimedia Telecommunications

Motivation

Passwords have long been a weak link in digital security, vulnerable to guessing, reuse and hacking. As cyber threats evolve, the need for more secure authentication methods has become urgent.

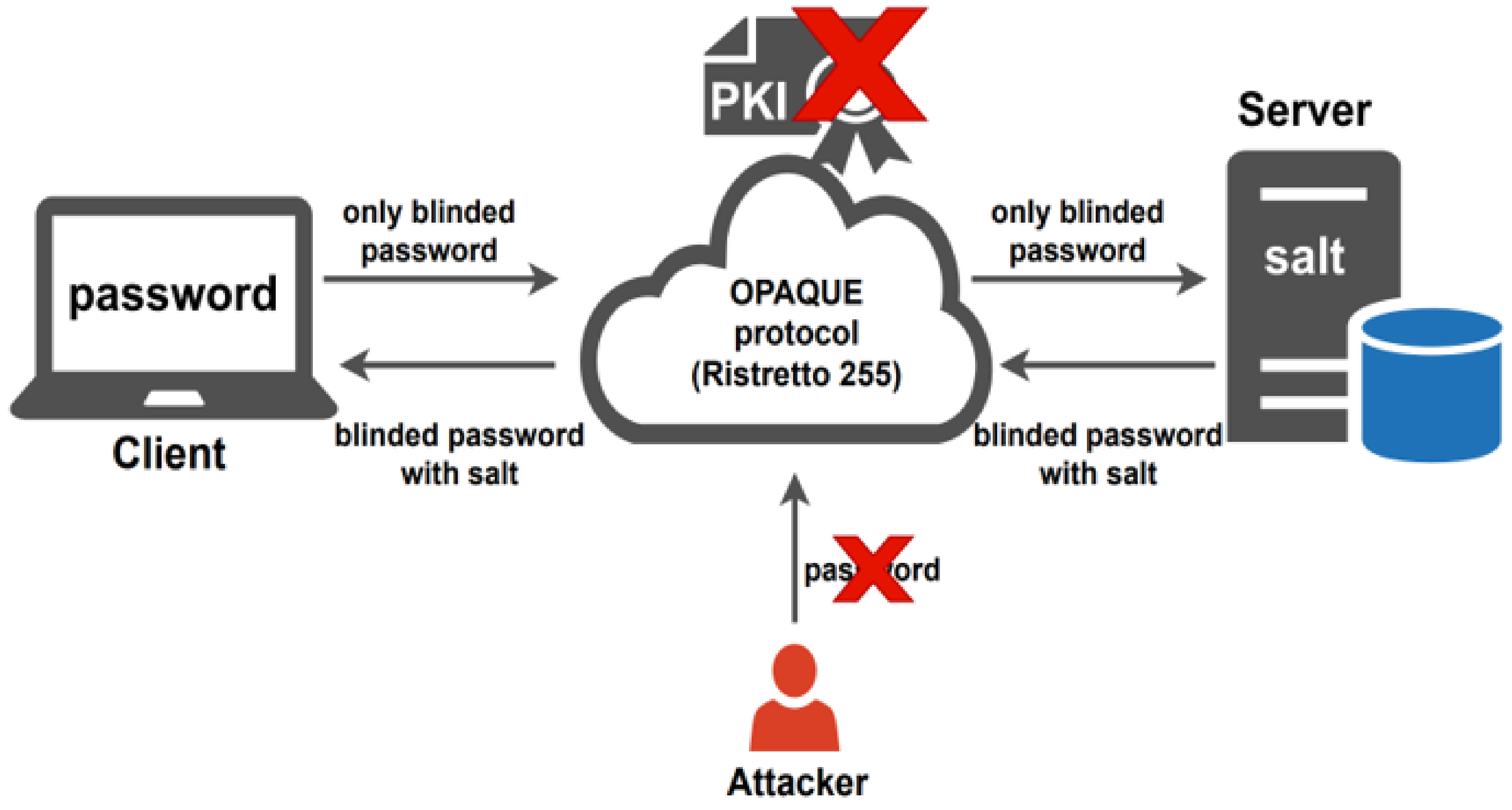
OPAQUE, an asymmetric password-authenticated key exchange (aPAKE) protocol, offers a breakthrough [1]. OPAQUE provides mutual authentication in a client-server setting. It allows users to prove their identities without revealing passwords, enhancing both privacy and security. OPAQUE does not store passwords on servers, defends against pre-computation attacks and supports forward secrecy.

Unlike traditional methods, OPAQUE is ideal for resource-constrained environments where the use of complex Public Key Infrastructure (PKI) would be impractical. Due to the absence of compact OPAQUE implementations targeted for microcontroller units (MCUs), we chose to develop our own implementation of the entire OPAQUE protocol.

We address the challenge of adapting OPAQUE for non-prime elliptic curves like Curve25519 [2] by implementing the Ristretto255 transformation (RG255) [3]. We provide an optimized, endian-agnostic OPAQUE implementation for embedded systems, specifically targeting the ARM Cortex-M4 core, contributing to a more secure and efficient client-server authentication protocol.

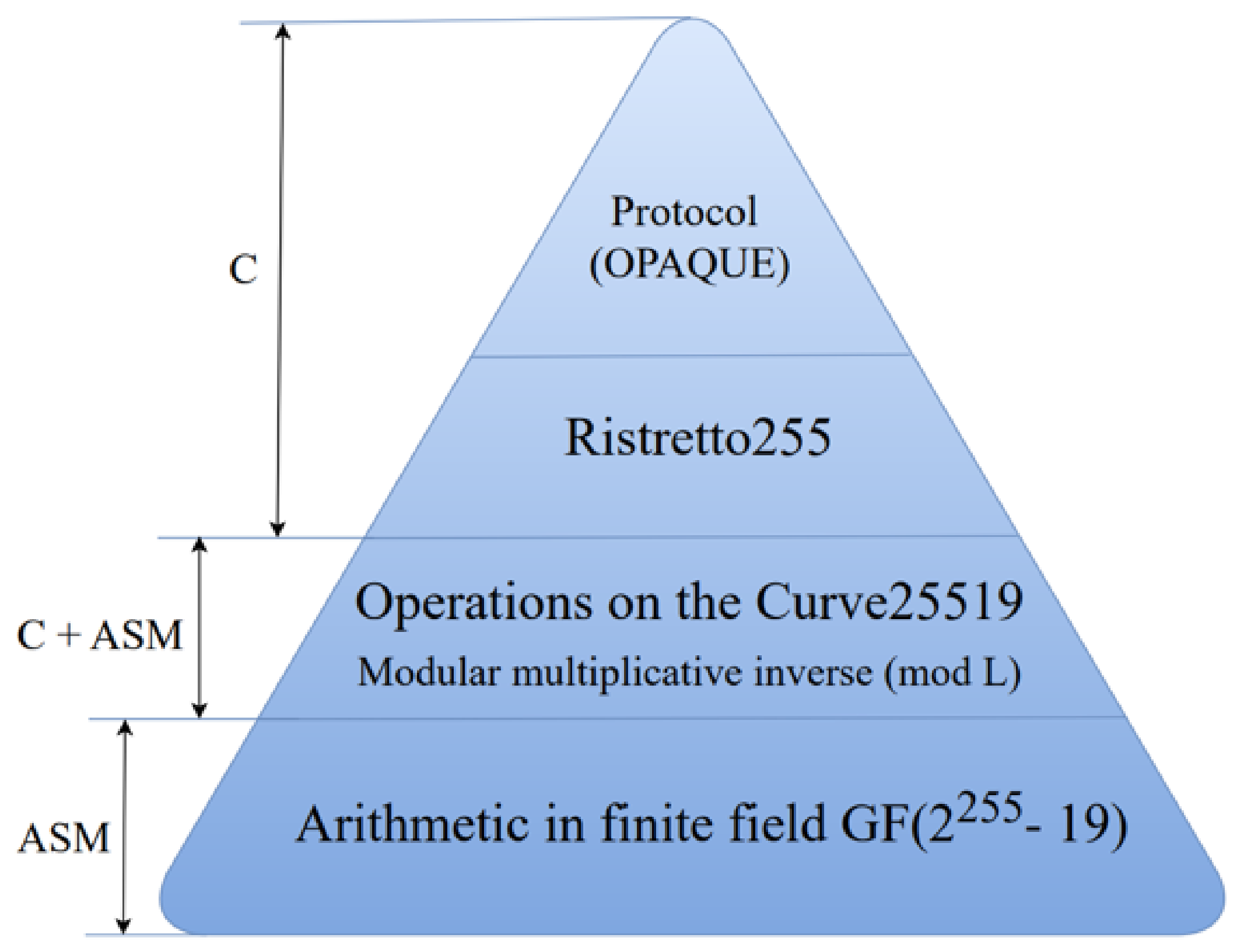
Client-server authentification without storing passwords?

OPAQUE is a modern password authentication protocol that avoids storing user passwords on the server. It ensures forward secrecy and prevents both the salt and user passwords from being exposed over insecure channels, making it highly resistant to breaches and offline dictionary attacks. While OPAQUE works without the need for a PKI [1], it can also integrate with PKI to further enhance and strengthen the overall authentication process.



Object of Our Optimization

Our implementation of the OPAQUE protocol, tailored for resource-constrained environments like embedded systems, utilizes several strategies and optimization techniques, divided into layers as depicted in the pyramid.


C
C + ASM
ASM
Protocol (OPAQUE)
Ristretto255
Operations on the Curve25519
Modular multiplicative inverse (mod L)
Arithmetic in finite field GF(2^255- 19)

Moving from the top to the bottom of the pyramid, Ristretto255, positioned in the second layer, represents a key aspect of our optimization efforts, constructed from simpler mathematical building blocks. The bottom layer of the pyramid illustrates arithmetic in the finite field $\mathbb{GF}(2^{255} - 19)$.

Finite field arithmetic, present in all upper layers, serves as a pivotal point for optimization. Consequently, our efforts concentrated on this lowest layer, as improvements here significantly impact the efficiency of all upper layers. Therefore, we utilized existing highly efficient ASM implementations specifically written for the ARM Cortex-M4 core and tweaked them to further enhance performance.

Highlights of our Optimization Techniques

- Portable Endian-Agnostic Code: Supports both little-endian and big-endian platforms, ensuring compatibility across diverse systems.
- Minimized Processor Stack Requirements: Optimized stack usage with shared local variables, reducing memory overhead.
- Advanced Mathematical and Coding Approaches: Explored and implemented the best mathematical techniques and coding strategies to enhance the performance of computationally intensive modular arithmetic operations.
- Highly Optimized ASM Routines: Incorporated existing highly optimized assembly routines for efficient $\mathbb{GF}(p)$ operations.
- Security: Preserved security with constant-time operations and secure wiping of local variables to prevent data leaks.

Results

By incorporating multiple optimization techniques, we have achieved a fast and compact client-server OPAQUE implementation while ensuring security. The following table summarizes the speed of each OPAQUE client-side function in terms of CPU cycles. These results are based on our optimization techniques, which include a combination of fast C code and highly efficient ASM operations. Measurements were taken on an STM32F4DISCOVERY development board with a little-endian ARM Cortex-M4 core running at 168 MHz, using the GCC ARM compiler. The table shows the client-side registration time (0.05 s) and login time (0.13 s), as the client-side is typically more resource-constrained than the server-side.

Setup	Registration Phase		Login Phase	
	Reg. Request [cycles]	Reg. Record [cycles]	KE1 [cycles]	KE3 [cycles]
ASM no wipe	2 189 663	5 041 000	4 234 479	13 944 523
ASM with wipe	2 548 919	5 893 381	5 893 381	16 210 575
Pure C no wipe	9 635 946	18 292 050	18 118 144	52 945 098

Conclusion

The thesis [4] presents a comprehensive study of the OPAQUE protocol, providing an explanation of its concepts and highlighting its potential for real-world applications, particularly with non-prime order groups like Curve25519. Thesis focuses on compact implementation of RG255 specifically optimized for the ARM Cortex-M4 core. The thesis covers the optimization of Ristretto255 computations in $\mathbb{GF}(2^{255} - 19)$, employing advanced techniques in ANSI C and Assembly [5], and integrating support for big-endian architecture. Experimental results confirm the effectiveness of our optimization techniques, which have significantly improved the performance of the OPAQUE protocol while maintaining its security. We aimed for a compact implementation with minimal memory usage. This research advances OPAQUE's practical use and sets the stage for further optimizations and integration into other cryptographic protocols. Additionally, our work on optimizing the RG255 for ARM Cortex-M4 was presented at an international conference [6].

References

[1] S. Jarecki, H. Krawczyk, and J. Xu, "OPAQUE: an asymmetric PAKE protocol secure against pre-computation attacks," in *Advances in Cryptology–EUROCRYPT 2018*. Springer, 2018, pp. 456–486.

[2] D. J. Bernstein, "Curve25519: new Diffie-Hellman speed records," in *Public Key Cryptography-PKC 2006*. Springer, 2006, pp. 207–228. [Online]. Available: https://cr.yp.to/ecdh/curve25519-20060209.pdf.

[3] M. Hamburg, "The ristretto group." [Online]. Available: https://ristretto.group/.

[4] P. Zeleňák and M. Drutarovský, "Compact opaque protocol implementation for embedded cryptographic applications," Master's thesis, Technical University of Košice, Faculty of Electrical Engineering and Informatics, Košice, 2024.

[5] L. Emil. X25519-cortex-m4. [Online]. Available: https://github.com/Emill/X25519-Cortex-M4

[6] E. Kupcova, P. Zelenak, M. Pleva, and M. Drutarovsky, "Optimization of ristretto255 group implementation for cortex-m4 based cryptographic applications," in *2024 34th International Conference Radioelektronika*, Žilina, Slovakia, April 17 2024, doi: https://10.1109/RADIOELEKTRONIKA61599.2024.10524097.