Towards a mobile eCard Client

J. Buchmann, A. Wiesmaier*, D. Hühnlein**, J. Braun, M. Horsch, F. Kiefer† and F. Strenzke††

* CASED Darmstadt ** ecsec GmbH Michelau † Technische Universität Darmstadt Darmstadt †† Flexsecure GmbH Darmstadt

Many future electronic identity cards will be equipped with a contact-less interface. Analysts expect that a significant proportion of future mobile phones support Near Field Communication (NFC) technology. Thus, it is a reasonable approach to use the cell phone as mobile smart card terminal, which in particular supports the Password Authenticated Connection Establishment (PACE) protocol to ensure user consent and to protect the wireless interface between the mobile phone and the smart card. Other than existing efficient PACE implementations on low power devices we present a platform independent solution using the Java Micro Edition (JavaME), which is supported by almost all modern mobile phones. Based on a straightforward PACE [1] implementation, we apply various optimizations realizable with existing JavaMe libraries to come up with a user friendly performance.

Since point multiplication is one of the biggest run-time consumers in the PACE protocol, we examine the elliptic curve arithmetic of the available Cryptographic Service Providers (CSP). Benchmarking [2] on PC and on the Nokia 6212 reveals the significant better performance of the FlexiProvider (FP) compared to Bouncy Castle which is the only other available CSP usable without changes. Thus, we use FP for our implementation. For further optimization we review the most common point multiplication algorithms and identify different possibilities for optimization, utilizing the advantages of the algorithms respectively. Merging different arithmetic operations of the PACE protocol allows us to utilize interleaved point multiplication and thus, reducing the total number of arithmetic operations. Furthermore we take advantage of different scalar lengths during the protocol.

Saving the domain-parameters of the used eID card at first contact allows for static and dynamic precomputations for subsequent program executions. Another optimization is changing the scheduling of the calculations needed during the protocol. Additionally, threading allows to use the time waiting for a response from the card or the user for further calculations. Further improvement is reached by avoiding heavy Java objects (which are given by the libraries) and using primitive data types and optimized data structures instead. We end up with a performance of 7.33 seconds on the first execution and 6.31 seconds when using the eID card repeatedly on the Nokia 6212, whereas the straightforward reference implementation achieves a total execution time of 9.5 seconds.

Additionally, we discuss potential side channel attacks and give advice on possible vulnerabilities. The future work discussion shows that there is more optimization potential when making changes to the existing CSPs.

All in all we succeeded in providing a platform independent efficient mobile PACE implementation [3], and also showed where and how more efficiency could be gained, thereby preparing the way for a mobile eCard client [4].

References

[1] M. Horsch. MobilePACE – Password Authenticated Connection Establishment implementation on mobile devices. Bachelor's Thesis, Department of Cryptography and Computer Algebra, Technische Universitt Darmstadt / CASED, September 2009.

[2] F. Kiefer. Effiziente Implementierung des PACE- und EAC-Protokolls fr mobile Gerte. Bachelor's Thesis, Department of Cryptography and Computer Algebra, Technische Universitt Darmstadt / CASED, July 2010.

[3] J. Buchmann, J. Braun, M. Horsch, D. Hühnlein, F. Kiefer, F. Strenzke, and A. Wiesmaier. An efficient PACE Implementation for mobile Devices. In preperation.

[4] J. Buchmann, J. Braun, M. Horsch, D. Hühnlein, T. Hühnlein, F. Kiefer and A. Wiesmaier. Mobile Authentisierung und Signatur. In preperation.