STSM Project Report

Cost Action IC1204 STSM

Host Institution: Digital Security Group, Radboud University Nijmegen, Ass. Prof. Peter Schwabe, Ass. Prof. Lejla Batina

Visitor: Pance Ribarski, Faculty of Computer Science and Engineering, Ss. Cyril and Methodius University in Skopje, pance.ribarski@finki.ukim.mk

Period: The proposed visiting period was from June 18, 2013 to July 19, 2013 (4 weeks).

Planned Research:

Cryptographic pairings on elliptic curves are maps of elements from two additive groups to a third multiplicative group. Cryptographic pairings are bilinear maps that were first introduced to cryptology as a tool to solve the discrete-logarithm problem on certain elliptic curves. The currently employed pairings are Weil, Tate, Ate and R-Ate pairings. The operations for computing these bilinear pairings are complex and seek for implementation of many algebraic operations in the three employed groups. In 2000 Joux described the first constructive application of pairings, a tripartite key-agreement protocol. Since then many cryptographic protocols have been developed that rely on pairings. Another line of research has greatly improved the performance of pairing computation.

The main focus of our research is the PandA Project. PandA represents a framework to test and benchmark implementations of bilinear pairings together with many algebraic operations needed for construction of protocols based on pairings on elliptic curves. This framework is designed to make it easy for researchers who optimize the performance of cryptographic pairings to include their results and for researchers in efficient pairing-based protocol design to make immediate use of the improvements in arithmetic performance. The project was incubated by Peter Schwabe and Michael Naehrig (now Microsoft Research, Redmond). Currently PandA doesn't have online documentiation and code repository. As soon as the framework together with our reference implementation of the API is mature and ready for public use, it will be published online and be placed in the public domain. The main aspects of PandA are:

- fast arithmetic in the underlying structures (elliptic curves and finite fields),
- fast multi-scalar multiplication in all three groups,
- hashing of arbitrary bit strings to elements of these structures,
- fast computation of products of pairings,

Furthermore, the algebraic operations are implemented in two modes: public-input and private-input versions. The private-input versions are equiped with side-channel-attack countermeasures. Side-channel attacks are attacks devised directly on the hardware, trying to gather information on working algorithms and sensitive data.

The PandA Project is based on the language C (but our high-speed implementation is using hand-optimized assembly for low-level finite-field arithmetic) and is easy to use in creation of complex protocols based on pairings. The PandA Project will have the latest and fastest

algorithms for computing pairings and other algebraic operations. This will be achieved by constant monitorng of the scientific work over the field of interest for PandA, and by doing benchmarking and looking at real protocols that would utilize PandA.

The visitor, Pance Ribarski, is doing this research as part of his PhD studies on a project for e-voting on constrained-hardware. Pairings on elliptic curves will have big impact on the speed and security of a new e-voting system and the possibility of mobile voting. The host institution has a group which works in the field of the proposed research and has valuable resources, both human and technological. The visit had a great impact in implementing the PandA Project and gaining significant knowledge and deliverables in the field of pairings on elliptic curves.

Layout of work done during the visit:

The scientific visit in Nijmegen produced good quality code for the PandA Project. The PandA Project had some starting code:

- the basic structure of PandA, implementations of basic arithmetics in the prime fields (addition, subtraction, multiplication and inversion);

- basic group arithmetics (mixed point addition where one of the points must be in affine coordinates, point doubling and double-and-add scalar multiplication);

- Bilinear pairing.

There was a lot more to be done for PandA. The basic arithmetics in the prime fields needed test cases and minor improvements. The three groups needed for cryptographic pairings were the second thing to implement. These groups needed optimized algorithms for point additions, point doublings and scalar multiplication. The missing methods were multi-scalar multiplication, hashing, packing and unpacking. Another supplemental data type are scalars with arithmetic modulo the group order, which are needed in various protocols.

PandA as a tool for the easy implementation of cryptographic pairings needed proof of concept for some protocols. We chose BLS signatures - “Short signatures from the Weil pairing” by Dan Boneh, Ben Lynn and Hovav Shachan. The implementation of BLS demonstrates how easy it is to implement pairing-based protocols in PandA and will set a new speed record for constant-time implementations of 128-bit secure BLS signatures.

Detailed work done during the visit:

Arithmetic operations in the underlying fields:

Addition, subtraction, multiplication, exponentiation, square-root computation, inversion, packing and unpacking (from and to byte arrays) for the prime field F_p. The work on this meant working with 256 bit numbers as operands using underlying four 64-bit limbs for each number. Automatic tests were made for checking the correctness of the arithmetic operations.

Addition, subtraction, multiplication, exponentiation, square-root computation, inversion, packing and unpacking for the prime field F_p^2. This prime field is an extension field and each element is a polynomial with two coefficients over F_p. The test suite was expanded with tests for F_p^2 arithmetic operations.

The implemented arithmetic operations for prime fieds F_p^6 and F_p^12 were as in F_p^2.

They are also Extension fields as F_p^2. The tests needed for correctness proofs were added to the automatic tests suite.

Arithmetic operations in the three groups and scalar representation:

The greater ammount of time spent in PandA implementation was the work for the group arithmetic. By definition, PandA project was supposed to have two types of arithmetic operations – public inputs, which should be fast without fear of leaking information, and side-channel-resistant operations, which should have mechanisms against side-channel attacks. Every implemented operation in the group and scalar arithmetic operations was implemented with both versions – public inputs and side-channel-resistant version.

The implemented algorithms for group 1 are:

- unified formula for point addition and point doubling,
- fixed-window scalar multiplication,
- sliding-window scalar multiplication,
- Strauss' algorithm for multi-scalar multiplication,
- Bos-Coster algorithm for multi-scalar multiplication,
- Hashing of strings into group 1.

The implemented algorithms for group 2 are:

- Signed window scalar multiplication
- Hashing of strings to group 2

Example of Cryptographic Pairing protocol:

There are many pairing protocols since 2000 Joux tripartite key-agreement protocol. We chose to implement BLS signatures protocol. This protocol is not complex, thus being a perfect candidate for proof of concept for PandA. Furthermore, cryptographic signatures are benchmarked within the eBACS benchmarking project run by Bernstein and Lange and we can submit the software to this project for public benchmarking. The BLS protocol, generally speaking, consists of three phases: key generation, signing, and verification. The key-generation phase chooses a random scalar *x* as private key. Then we derive the public key *p* by scalar multiplication of the *base point* in group 2 with the private key *x*. The signing phase requires hashing of the message to an element *msg_p* of group 1. Then, the signing represents scalar multiplication of the point *msg_p* with the private key *x.* The result *s* is the signature of the message. The verification requires computing two pairings: *e1* is the pairing of public key *p* and the hashed message *msg_p; e2* is the pairing between the *base point* in group 2 and the generated signature *s.* If the pairings *e1* and *e2* are equal then the signature is verified. Our test implementation of BLS consists of merely six lines of code using the PandA API. This is a clear example of the usability and simplicity of PandA for pairing-based protocols.

Future work:

As explained in the example of BLS signatures, PandA is a working project for Pairings Cryptography protocols. But there are more algorithms that need implementation. Missing algorithms for group 2: Sliding window scalar multiplication, Strauss multi-scalar multiplication and Boss-Coster multi-scalar multiplication. Group 3 currently does not have these algorithms implemented, that is also work to be done because various protocols (for example, somewhat homomorphic encryption) also need efficient arithmetic in group 3.

There are other non-functional details that need attention for PandA. Better test suites and build framework need to be devised in order of publicizing PandA. There is an idea of integrating PandA with the eBACS benchmarking tool SUPERCOP.

Also, listed as future work, is implementation of more complex protocols for pairing-based cryptography. The idea is to work on e-voting protocols which utilize cryptographic pairings. This will most likely require extending the high-performance implementations PandA to software optimized for small embedded processors (currently we are focussing on large desktop and server processors). Note however, that implementation of more complex protocols *using* PandA and further optimization and extension of pairing software *within* PandA can process independently, with the PandA API bridging between the two.

Presentation of PandA Project:

We are currently in the process of writing a paper presending PandA. This paper will be submited to “The 6th International Conference on Pairing-Based Cryptography (Pairing 2013)” which will be held in Beijing on 22-24 November.

PandA is planned to be presented in the rump session of the “17th Workshop on Elliptic Curve Cryptography” which will be held in Leuven on 16-18 September. This is a nice opportunity to present the project to scientist in the field of Elliptic Curves and show the impact the PandA can make in pairing-based cryptography.