Diploma Work: Development of a Compiler for Zero-Knowledge Proof-of-Knowledge Protocols

The goal of this diploma work is to develop a language for the specification of so called "zero-knowledge proofs of knowledge of a preimage of a homomorphism" protocols and a corresponding compiler which translates protocol specifications into JAVA source code which realizes the protocol.

"Zero-knowledge proofs of knowledge of a preimage of a homomorphism" are protocols between two parties called prover and verifier. Informally, these two-party protocols have the following properties: a function $f()$ (group homomorphism) and a y are given to both the prover and verifier. Additionally the prover is given a secret value x such that $y = f(x)$. At the end of a successful protocol execution it holds that

- the verifier can be assured, that the prover indeed knows the secret corresponding to y with respect to f,
- the prover can be assured, that the verifier did not learn anything about the secret value x.

Several basic protocols with these properties do exist. Furthermore, by combining such basic protocols knowledge of relations among the secrets can be proved. For instance, the prover can prove that, given y_1, y_2, $f_1()$, $f_2()$, and two integers a and b, it knows secret values x_1 and x_2 such that the relation $ax_1 + bx_2 = 0$ is fulfilled (whereas the verifier does not gain knowledge about the values of x_1 and x_2). Such protocols are fascinating from a purely theoretical point of view and considerable theoretical research was conducted on the topic. From a practical point of view these protocols play an important role as sub-protocols in higher level protocols, such as novel anonymous public key infrastructures and secure multi-party computations.

Currently such systems and protocols are being developed at the IBM Zurich Lab (for an overview please refer to http://www.zurich.ibm.com/security/idemix). Hence the aim arises not only to consider these protocols from a theoretical point of view but also to actually implement software systems that make use of zero-knowledge proofs of knowledge. As a matter of fact, for someone having certain cryptographic knowledge, given some function $g()$ it is fairly easy to choose which protocol of many possible ones is appropriate to prove knowledge of secret with respect to $g()$. However, it is quite *time consuming* and *error prone* to actually generate a description (implementation) of the chosen protocol for instance in the JAVA programming language. This observation constitutes the motivation of the diploma work, which has the objective to automate the translation process from a protocol specification to the protocol description / implementation. More precisely, the goal of the diploma-work is to

- create in collaboration with a cryptographer a *protocol specification language* for zero-knowledge proofs of knowledge of a preimage of a homomorphism..
- design and implement a *compiler* which translates protocol specifications into a target language, e.g., JAVA.

We are looking for students with theoretical and practical knowledge in compiler design / implementation and good JAVA programming skills. While no cryptographic knowledge is required, the student should be interested in learning some of the cryptographic theory underlying the diploma work.

For more information please contact: IBM Zurich Research Lab, Endre Bangerter, Saeumerstr. 4, 8803 Rueschlikon, Switzerland. e-mail: eba@zurich.ibm.com