Advanced cryptography: Pairing-based cryptography
winter term 2012/13

DANIEL LOEBENBERGER AND MICHAEL NÜSKEN

6. Exercise sheet
Hand in solutions until Monday, 10 December 2012, 23:59:59

Exercise 6.1 (Hashing into a curve). (15 points)

Consider an elliptic curve over $\mathbb{F}_p$ for some prime p of, say, length 160bit. Furthermore consider a hash-function h (such as SHA-3 with 256 bit output) that produces for any message m a 256-bit hash-value.

(i) Consider the following hash function $h\colon \{0,1\}^* \to E(\mathbb{F}_p)$ that first computes $h(m)$, truncates the result to 160 bit and increments the value until a valid x-coordinate is found. The next bit of the $h(m)$ value is then used to determine which of the two y coordinates to use. Discuss this straightforward approach. What can you say about the security of the procedure with respect to collision resistance and pre-image resistance? [5]

(ii) In fact, the problem of hashing into an elliptic curve is an active research area. One good article on the topic can be found at [10]

`http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.215.5920`.

Read the article and give a short survey on state of the art methods. Focus in your exposition on the methods used, as well as the security aspects.

Exercise 6.2 (A missing proof). (5 points)

Prove that if the BDH problem is broken then there is an INDSK attacker for the Sakai, Ohgishi & Kasahara key distribution system. [5]

Exercise 6.3 (A generalized proof). (10+2 points)

The Sakai, Ohgishi & Kasahara key distribution system employs a pairing $e\colon G_1 \times G_2 \to G_T$. However, the proof of security just worked in the symmetric setting with $G_1 = G_2$. Improve it such that you can show the same in the asymmetric setting, i.e. in the case $G_1 \neq G_2$. [10]

Do you trust the proof? [+2]

Exercise 6.4 (Do some research). (0 points)

Try to find out whether we can adapt the proof working for the Sakai, Ohgishi & Kasahara key distribution system in the Dupont & Enge setting assuming the DBDH problem is hard. [+0]