4.2 Diffie-Hellman Key Exchange

We treat some exemplary applications that provide astonishingly elegant solutions for seemingly unsolvable problems under the discrete logarithm assumption.

Imagine A (Alice) and B (Bob) want to exchange a key for a symmetric cipher. In 1976 Diffie and Hellman proposed the following protocol whose security relies on the dicrete logarithm assumption:

1. A and B (publicly) agree on a prime p and a primitive element $a \bmod p$.
2. A generates a random integer x, computes $u = a^x \bmod p$, and sends u to B.
3. B generates a random integer y, computes $v = a^y \bmod p$, and sends v to A.
4. A computes $k = v^x \bmod p$, and B computes $k = u^y \bmod p$.

Now A and B share a secret k that may be used as key. The fact that A and B compute the same key k lies in the equation

$$v^x \equiv a^{xy} \equiv u^y \pmod{p}.$$

An eavesdropper can intercept the values p, a, u, and v. But this doesn't enable her to efficiently compute k, or x, or y.

This protocol realizes a kind of hybrid encryption. A difference with a "proper" asymmetric cipher concerns the need for synchronization between A and B, preventing spontaneous messages (for example by e-mail that follows an asynchroneous protocol).

An attacker who is able to efficiently compute discrete logarithms is also able to efficiently break the Diffie-Hellman protocol. It is unknown whther the converse also holds.

The British Secret Service CESC knew the procedure already in 1974 but of course kept it secret.

Here is a mathematical model for a somewhat more abstract protocol:

1. A and B (publicly) agree on a set X, an element $a \in X$, and a commutative subsemigroup $H \subseteq \mathrm{Map}(X, X)$.
2. A chooses a random map $\varphi_A \in H$, computes $u = \varphi_A(a)$, and sends u to B.
3. B chooses a random map $\varphi_B \in H$, computes $v = \varphi_B(a)$, and sends v to A.
4. A computes $\varphi_A(v)$, and B computes $\varphi_B(u)$.

Then A and B share the secret value

$$k = \varphi_A(v) = \varphi_A(\varphi_B(a)) = \varphi_B(\varphi_A(a)) = \varphi_B(u)$$

and may use it as key for their secret communication—at least if an attacker has no method to derive φ_A, φ_B, or k from the entities X, a, u, and v she knows or intercepts.

For the adaption of this protocol to elliptic curves an even more abstract scenario is useful that is visualized by a commutative diagram as follows: