7 Diffie-Hellman, ElGamal, and a Bit of History

7.1 Diffie-Hellman Key Exchange

A great surge of academic interest in modern cryptography started with the work of Diffie, Hellman, and Merkle, and the publication of "New Directions in Cryptography" by Diffie and Hellman [DH76]. In this work, Diffie and Hellman proposed the idea of public-key encryption and digital signatures. Although they didn't have an implementation of public-key encryption, they did suggest something close, called "key agreement."

Here is the idea. Suppose there is a fixed prime p and generator g of $\mathbb{Z}_p^*$ known to everyone. Alice and Bob want to agree on a secret they can both use for some symmetric encryption scheme. To do so, Alice selects a random $a \in \mathbb{Z}_p^*$ and sends $g^a \bmod p$ to Bob. Bob similarly selects a random $b \in \mathbb{Z}_p^*$ and sends $g^b \bmod p$ to Alice. Now Alice can compute $K = g^{ab}$ by raising g^b to the power a, and Bob similarly can compute K by raising g^a to the power b. It is believed that g^{ab} is hard to compute from just g, g^a and g^b. More formally, this is known as the Computational Diffie-Hellman Assumption.

Assumption 1. For any poly-time algorithm A, there exists a negligible function η such that, if you generate random k-bit prime p and its generator g, and select a random $a, b \in \mathbb{Z}_p^*$, $\Pr[A(p, g, g^a \bmod p, g^b \bmod p) = (g^{ab} \bmod p)] \leq \eta(k)$.

Note that if p and g are not known to both parties in advance, Alice can simply send both to Bob together with g^a.

7.2 A Bit More History

In 1977, the RSA cryptosystem [RSA78] appeared in Scientific American, helping generate public interest in the subject.

Until 1976, research in cryptography was mostly done in classified research labs, such as the National Security Agency in the United States, for military and intelligence purposes. Documents declassified by the UK in the late 1990s and now available on the web [Ell87] showed that public-key cryptography in general, and Diffie-Hellman and RSA specifically, were discovered in the classified community before their discovery in academia. Specifically, in 1970, James H. Ellis [Ell70] proposed the idea of public-key cryptography, which he termed "non-secret encryption"; in 1973, Clifford C. Cocks [Coc73] proposed RSA (although Cocks suggested using specific public exponent n, equal to the modulus, rather than a more general public exponent); and in 1974, Malcolm J. Williamson [Wil74, Wil76] proposed what we know as Diffie-Hellman. It's worth noting that the discoveries of RSA and Diffie-Hellman occurred in reverse order in the classified community, and that neither preceded the academic discoveries by more than a few years. It seems (according to what we know) that there wasn't much interest in public-key encryption in the military and intelligence community. One possible reason is that with rigid command structures such as those in the military, it is easy enough to establish shared secret keys (public-key ideas are of great help when people who have never seen each other before want to talk; this doesn't happen too much in the military). The second commonly cited reason is that the state of computers in the 1970s did not allow for such expensive operations as modular exponentiation to be easily carried out "in the field."

7.3 Man-in-the-middle attack against Diffie-Hellman

Imagine now that an adversary Eli is capable of not only intercepting messages between Alice and Bob, but also stopping them and substituting his own messages instead. Then Eli can do the following: pick his own random $e \in \mathbb{Z}_p^*$, and compute $g^e \bmod p$. Then intercept g^a that Alice sends to Bob, and substitute g^e instead. Note that Bob doesn't notice any difference (because, after all, both g^a and g^e are random elements of $\mathbb{Z}_p^*$), and dutifully replies with g^b. Eli intercepts g^b, and sends g^e to Alice instead. This way, Alice ends up thinking that she is sharing $K_1 = g^{ea}$ with Bob, while Bob ends up thinking that he is sharing $K_2 = g^{eb}$ with Alice. Note that, in fact, they are both sharing a key with Eli, who can compute g^{ea} and g^{eb}. Now whenever Bob tries to send something to Alice, he'll presumably encrypt (and/or authenticate) it using K_2. Eli can intercept it, decrypt with K_2, reencrypt with K_1, and send it on to Alice. So Bob and Alice will never realize they aren't sharing a key with each other.

This is known as "man-in-the-middle" attack, and is just one of the reasons why key agreement is a difficult problem. In fact, satisfactory formal definitions for key agreement took about a decade and a half longer to appear than definitions for encryption and signature. We will not study key agreement in this class. We will, however, use Diffie-Hellman below.

7.4 ElGamal Encryption

Taher ElGamal [ElG85] proposed the following way to make Diffie-Hellman into an encryption scheme. Alice publishes $p, g, g^a \bmod p$, as a public key, and keeps a as the secret key. To encrypt a message $m \in \mathbb{Z}_p^*$, Bob picks $b \in \mathbb{Z}_p^*$ at random, computes $g^b \bmod p$, $K = g^{ab} \bmod p$, and $c = mK \bmod p$, and outputs $(c, g^b \bmod p)$. To decrypt, Alice computes K using g^b and a, and recovers m from mK by dividing.

The scheme as described above is not semantically secure, because there exists a distinguisher D with good probability of success. Here is how D works: it outputs two messages m_0 and m_1, such that $m_0 \in QR_p$ and $m_1 \notin QR_p$. Then, upon receiving the ciphertext $(c, g^b \bmod p)$, D checks if $c \in QR_p$ (by checking whether $c^{(p-1)/2} \bmod p$ is 1 or -1). If so, it outputs 1; else it outputs 0. Note that $K \in QR_p$ if and only if ab is even, i.e., with probability 3/4. Therefore, if $m \in QR_p$, then $mK \in QR_p$ with probability 3/4; if $m \notin QR_p$, then $mK \in QR_p$ with probability 1/4 (because a non-square times a non-square is a square, but a non-square times a square is a non-square). Hence, the difference of the probabilities of D's output being 1 on encryption of m_0 and encryption of m_1 is $3/4 - 1/4 = 1/2$, which is not negligible.

However, ElGamal scheme can be fixed if we restrict our attention not the entire group $\mathbb{Z}_p^*$, but rather to the subgroup of squares QR_p. If this subgroup is of prime order (i.e., if $(p-1)/2$ is a prime), then p is often called a *safe* prime (and $(p-1)/2$ a *Sophie Germain* prime). Then the following assumption is believed to hold.

Assumption 2. For any poly-time algorithm A, there exists a negligible function η such that, if you generate random k-bit safe prime $p = 2q+1$ for prime q, and select a random generator g of QR_p, and random integers a, b and c between 1 and q,

$$\begin{aligned}&|\Pr[A(p, g, g^a \bmod p, g^b \bmod p, g^{ab} \bmod p) = 1] - \\ &\quad \Pr[A(p, g, g^a \bmod p, g^b \bmod p, g^c \bmod p) = 1]| \quad \leq \quad \eta(k).\end{aligned}$$

This is known as the Decision Diffie-Hellman (DDH) assumption, because it states that it's hard to decide whether you got g^{ab} or g^c for a random c. Note that this is a much stronger assumption than Computational Diffie-Hellman (CDH): CDH states that it's hard to compute g^{ab}, while DDH states that not only is it hard to compute, it actually looks random. There are many who are uncomfortable with such a strong assumption.

Let us now reformulate ElGamal encryption to take advantage of DDH. Alice publishes as her public key $p = 2q + 1$, where q is prime; g of order q, which is a generator of QR_p; and $g^a \bmod p$, for a random a between 1 and q. She keeps a as her secret key. To encrypt a message m, $1 \leq m \leq q$, Bob picks b, $1 \leq b \leq q$ at random, computes $g^b \bmod p$, $K = g^{ab} \bmod p$, and $c = m^2K \bmod p$ and outputs $(c, g^b \bmod p)$. To decrypt, Alice computes K using g^b and a, and recovers m^2 from m^2K by dividing. She then finds m by taking a square root (note that there are two square roots, but one is greater than $q = (p-1)/2$, so she knows which one is m).

Theorem 1. *The above cryptosystem is polynomially secure under the DDH assumption.*

The proof, which is not presented in full detail here, is by hybrid argument: one proves that encryption of any message m is indistinguishable from a random pair (g^c, g^b). This follows easily from the DDH assumption. Therefore, encryptions of m_0 and m_1 are indistinguishable.

References

[Coc73] Clifford C. Cocks. A note on non-secret encryption, 1973. Available from `http://www.cesg.gov.uk/publications/index.htm`.

[DH76] Whitfield Diffie and Martin E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, IT-22(6):644–654, 1976.

[ElG85] Taher ElGamal. A public-key cryptosystem and a signature scheme based on the discrete logarithm. *IEEE Transactions of Information Theory*, 31(4):469–472, 1985.

[Ell70] James H. Ellis. The possibility of non-secret encryption, 1970. Available from `http://www.cesg.gov.uk/publications/index.htm`.

[Ell87] James H. Ellis. The story of non-secret encryption, 1987. Available from `http://www.cesg.gov.uk/publications/index.htm`.

[RSA78] Ronald L. Rivest, Adi Shamir, and Leonard M. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, February 1978.

[Wil74] Malcolm J. Williamson. Non-secret encryption using a finite field, 1974. Available from `http://www.cesg.gov.uk/publications/index.htm`.

[Wil76] Malcolm J. Williamson. Thoughts on cheaper non-secret encryption, 1976. Available from `http://www.cesg.gov.uk/publications/index.htm`.