CS 346 Class Notes

Mark Lindberg

Apr 18, 2016

This Time:
Chapter 10: Key management and the public-key revolution.
Central question: How can we establish a secret key for communication between two parties?
10.3 Diffie-Hellman key exchange protocol.
10.2 Key distribution centers (KDCs). For example, in a company environment.
Alice wants to communicate with Bob.
Naive approach:

- Alice informs the KDC.
- The KDC determines the session key k.
- The KDC sends $\mathsf{Enc}_{k_A}(k)$ to Alice, and $\mathsf{Enc}_{k_B}(k)$ to Bob, where k_A is Alice's private key, known only to her and the KDC, and likewise with k_B and Bob.

Needham-Schroeder Variant:

- KDC sends $\mathsf{Enc}_{k_A}(k)$ and ticket=$\mathsf{Enc}_{k_B}(k)$ to Alice.
- Alice sends the ticket to Bob to initiate the session.

Rest of the day: 10.3! Diffie-Helman.
Generic key exchange protocol:

- Alice and Bob each get the same security parameter n. (in unary)
- Alice outputs k_A; Bob outputs k_B, each of which are n-bit strings.

Security of a key exchange protocol Π against an eavesdropper.
$\mathsf{KE}^{EAV}_{\mathcal{A},\Pi}(n)$:

1. Alice, Bob get 1^n, and run Π. Produces trace T. Key $k = k_A = k_B$.
2. Pick a random bit b. If $b = 0$, set $\hat{k} = k$. Otherwise, set $\hat{k}$ to a uniformly random n-bit value.

3. $\mathcal{A}$ is given T and k. $\mathcal{A}$ outputs $b' \in \{0, 1\}$.

4. $\mathcal{A}$ succeeds iff $b = b'$.

Security. Definition 10.1. Π is secure in the presence of an eavesdropper if $\forall$ PPT adversaries $\mathcal{A}$, $\Pr[\mathsf{KE}^{EAV}_{\mathcal{A},\Pi}(n) = 1] \leq \frac{1}{2} + \mathtt{negl}(n)$.

Thing.

Let $\mathbb{G}$ be a group generation algorithm, with input a^n.

It produces (G, q, g), where

1. G is a (description of a) cyclic group.

2. q is the order of G and $||q|| = n$. (q is an n-bit number.)

3. g is a generator of G.

Diffie-Hellman key exchange protocol. I'm glad I have new glasses.

1. Alice runs $\mathbb{G}(1^n)$ to get (G, q, g).

2. Alice generates a uniform random $x \in \mathbb{Z}_q$. and computes $k_A = g^x$. (Therefore, k_A is a uniform random element of G.)

3. Alice sends (G, q, g, k_A) to Bob.

4. Bob generates a uniform random $y = in\mathbb{Z}_q$ and computes $k_B = g^y$. Bob sends k_B to Alice, and outputs $k_A^y = g^{xy}$

5. Alice outputs $k_B^x = g^{xy}$.

This output is their shared, private key.

Stronger assumptions than the discrete log assumption are needed for Diffie-Hellman to be secure.

Computational Diffie-Hellman assumption (CDH).

Given h_1, h_2, uniformly random group elements, compute $DH_g(h_1, h_2) := g^{[\log_g(h_1) \cdot \log_g(h_2)]}$.

Due to our security definition being based on indistinguishability, we need a stronger assumption: Decisional Diffie-Hellman!

Definition 8.63. DDH hard relative to $\mathbb{G}$ if $|\Pr[\mathcal{A}(G, q, g, g^x, g^y, g^z) = 1] - Pr[\mathcal{A}(G, q, g, g^x, g^y, g^{xy}) = 1]| \leq$ $\mathtt{negl}(n)$.

The probabilities are with respect to

1. Randomness of $\mathbb{G}$.

2. Uniformly random choice of x, y, z in $\mathbb{Z}_q$.

Almost exactly what we need to show security, as in definition 10.1.

Technicality: As described, the output of the protocol is not an n-bit string, but a random group element. In practice, we use hash functions to map group elements to n-bit strings. Security is shown with respect to the random group element.

The security of the Diffie-Hellman key exchange protocol with respect to the modified definition 10.1.

Let $\mathcal{A}$ be an arbitrary PPT adversary.

$$\begin{aligned}
\Pr[\mathsf{KE}^{\hat{E}AV}_{\mathcal{A},\Pi}(n) = 1] &= \frac{1}{2}[\Pr(\mathsf{KE}^{\hat{E}AV}_{\mathcal{A},\Pi}(n) = 1 \mid b = 0) + \Pr(\mathsf{KE}^{\hat{E}AV}_{\mathcal{A},\Pi}(n) = 1 \mid b = 1)] \\
&= \frac{1}{2}\Pr[\mathcal{A}(G, q, g, g^x, g^p, g^{xy}) = 1] + \frac{1}{2}\Pr[\mathcal{A}(G, q, g, g^x, g^y, g^z)] \\
&\leq \frac{1}{2} + \frac{1}{2}[\Pr(\mathcal{A}(G, q, g, g^x, g^y, g^z) = 1) - \Pr[\mathcal{A}(G, q, g, g^x, g^y, g^{xy} = 1]]
\end{aligned}$$