Public Key Crypto: DH

Lecture 06

The Diffie-Helman Key Exchange

Alice and Bob will share a secret s.

1. Alice finds a (p, g), p of length n, g gen for $\mathbb{Z}_p$. Arith mod p.
2. Alice sends (p, g) to Bob in the clear (Eve can see it).
3. Alice picks random $a \in \{\frac{p}{3}, \ldots, \frac{2p}{3}\}$. Alice computes g^a and sends it to Bob in the clear (Eve can see it).
4. Bob picks random $b \in \{\frac{p}{3}, \ldots, \frac{2p}{3}\}$. Bob computes g^b and sends it to Alice in the clear (Eve can see it).
5. Alice computes $(g^b)^a = g^{ab}$.
6. Bob computes $(g^a)^b = g^{ab}$.
7. g^{ab} is the shared secret.

The Diffie-Helman Key Exchange

Alice and Bob will share a secret s.

1. Alice finds a (p, g), p of length n, g gen for $\mathbb{Z}_p$. Arith mod p.
2. Alice sends (p, g) to Bob in the clear (Eve can see it).
3. Alice picks random $a \in \{\frac{p}{3}, \ldots, \frac{2p}{3}\}$. Alice computes g^a and sends it to Bob in the clear (Eve can see it).
4. Bob picks random $b \in \{\frac{p}{3}, \ldots, \frac{2p}{3}\}$. Bob computes g^b and sends it to Alice in the clear (Eve can see it).
5. Alice computes $(g^b)^a = g^{ab}$.
6. Bob computes $(g^a)^b = g^{ab}$.
7. g^{ab} is the shared secret.

PRO: Alice and Bob can execute the protocol easily.

The Diffie-Helman Key Exchange

Alice and Bob will share a secret s.

1. Alice finds a (p, g), p of length n, g gen for $\mathbb{Z}_p$. Arith mod p.
2. Alice sends (p, g) to Bob in the clear (Eve can see it).
3. Alice picks random $a \in \{\frac{p}{3}, \ldots, \frac{2p}{3}\}$. Alice computes g^a and sends it to Bob in the clear (Eve can see it).
4. Bob picks random $b \in \{\frac{p}{3}, \ldots, \frac{2p}{3}\}$. Bob computes g^b and sends it to Alice in the clear (Eve can see it).
5. Alice computes $(g^b)^a = g^{ab}$.
6. Bob computes $(g^a)^b = g^{ab}$.
7. g^{ab} is the shared secret.

PRO: Alice and Bob can execute the protocol easily.
Biggest PRO: Alice and Bob never had to meet!

The Diffie-Helman Key Exchange

Alice and Bob will share a secret s.

1. Alice finds a (p, g), p of length n, g gen for $\mathbb{Z}_p$. Arith mod p.
2. Alice sends (p, g) to Bob in the clear (Eve can see it).
3. Alice picks random $a \in \{\frac{p}{3}, \ldots, \frac{2p}{3}\}$. Alice computes g^a and sends it to Bob in the clear (Eve can see it).
4. Bob picks random $b \in \{\frac{p}{3}, \ldots, \frac{2p}{3}\}$. Bob computes g^b and sends it to Alice in the clear (Eve can see it).
5. Alice computes $(g^b)^a = g^{ab}$.
6. Bob computes $(g^a)^b = g^{ab}$.
7. g^{ab} is the shared secret.

PRO: Alice and Bob can execute the protocol easily.
Biggest PRO: Alice and Bob never had to meet!
Question: Can Eve find out s?

What Do We Really Know about Diffie Helman?

If Eve can compute Discrete Log quickly then she can crack DH:

1. Eve sees g^a, g^b.
2. Eve computes Discrete Log to find a, b.
3. Eve computes $g^{ab} \pmod{p}$.

If Discrete Log Easy then DH is crackable

What Do We Really Know about Diffie Helman?

If Eve can compute Discrete Log quickly then she can crack DH:

1. Eve sees g^a, g^b.
2. Eve computes Discrete Log to find a, b.
3. Eve computes $g^{ab} \pmod{p}$.

If Discrete Log Easy then DH is crackable

What about converse?
UNKNOWN TO SCIENCE.

Hardness Assumption

Definition

Let f be the following function:

Input: p, g, g^a, g^b (note that a, b are not the input)

Outputs: g^{ab}.

Hardness assumption: f is hard to compute.

We may later show how to prove, assuming the hardness assumption, that DH is hard to crack. But this proof will depend on a model of security that Eve is not obliged to work in.

What Could be True?

The following are all possible:

1) Discrete Log is easy. Then DH is crackable.

2) DL is hard but Hardness Assumption is false. Then DH is crackable even though DL is hard!!

3) DL is hard, Hardness Assumption is true, but DH is crackable by outside-the-box thinking. Timing Attacks. This would force us to rethink our model of security.

4) DL is hard, Hardness Assumption is true, and DH remains uncracked for years. Increases our confidence but ....

Item 4 is current state with some caveats: Do Alice and Bob use it properly? Do they have large enough parameters? What is Eve's computing power?

Attacks on DH

Paper: Imperfect Forward Secrecy: How DH Fails in Practice.
Breaks DH using the following

1) Alice and Bob p, g for a long time. Eve can preprocess.

2) Amortize: Solve many DL's easier per-problem than just one.

3) State-of-the-art Number Theory is just enough.

4) If p is not a safe prime then can use this to help crack.

Non-Caveat: The paper is not hypothetical. They really used there method to crack real systems that are really in use.

Attacks on DH

Paper: Imperfect Forward Secrecy: How DH Fails in Practice.
Breaks DH using the following

1) Alice and Bob p, g for a long time. Eve can preprocess.

2) Amortize: Solve many DL's easier per-problem than just one.

3) State-of-the-art Number Theory is just enough.

4) If p is not a safe prime then can use this to help crack.

Non-Caveat: The paper is not hypothetical. They really used there method to crack real systems that are really in use.
The authors have not been heard from since!

Attacks on DH

Paper: Imperfect Forward Secrecy: How DH Fails in Practice.
Breaks DH using the following

1) Alice and Bob p, g for a long time. Eve can preprocess.

2) Amortize: Solve many DL's easier per-problem than just one.

3) State-of-the-art Number Theory is just enough.

4) If p is not a safe prime then can use this to help crack.

Non-Caveat: The paper is not hypothetical. They really used there method to crack real systems that are really in use.
The authors have not been heard from since!
I am kidding about that, but NOT about them cracking real systems.

Recall Diffie-Helman

1. Alice and Bob end up sharing a secret.
2. p, g are public keys.
3. Under a hardness assumption Eve does not know the secret.
4. The secret is *not* in Alice or Bob's control

DH cannot be used for the following:

Alice takes the message Lets do our Math/CMSC 456 HW on time this week for a change encrypt it, send it to Bob, and Bob Decrypts it.

We describe the ElGamal Public Key Encryption Scheme where Alice and Bob can encrypt and decrypt under a hardness assumption.

ElGamal is DH with a Twist

1. Alice and Bob do Diffie Helman.
2. Alice and Bob share secret $s = g^{ab}$.
3. Alice and Bob compute $(g^{ab})^{-1} \pmod{p}$.
4. To send m, Alice sends $c = mg^{ab}$
5. To decrypt, Bob computes $c(g^{ab})^{-1} \equiv mg^{ab}(g^{ab})^{-1} \equiv m$

We omit discussion of Hardness assumption (HW)