BILL TAPE LECTURE

Diffie-Helman Key Exchange

Summary of Where We Are

Summary of Where We Are

1. Finding primes p such that $p-1=2q$, q a prime, EASY

Summary of Where We Are

1. Finding primes p such that $p - 1 = 2q$, q a prime, EASY
2. Given such a p, finding generator g, EASY.

Summary of Where We Are

1. Finding primes p such that $p - 1 = 2q$, q a prime, EASY
2. Given such a p, finding generator g, EASY.
3. Given such a p, finding generator $g \in \{\frac{p}{3}, \ldots, \frac{2p}{3}\}$ EASY.

Summary of Where We Are

1. Finding primes p such that $p-1=2q$, q a prime, EASY
2. Given such a p, finding generator g, EASY.
3. Given such a p, finding generator $g \in \{\frac{p}{3}, \ldots, \frac{2p}{3}\}$ EASY.
4. Given p, g, a finding $g^a \pmod{p}$ EASY.

Summary of Where We Are

1. Finding primes p such that $p-1=2q$, q a prime, EASY
2. Given such a p, finding generator g, EASY.
3. Given such a p, finding generator $g \in \{\frac{p}{3}, \ldots, \frac{2p}{3}\}$ EASY.
4. Given p, g, a finding $g^a \pmod{p}$ EASY.
5. The following problem thought to be hard:
   Input: prime p, generator $g \in \{\frac{p}{3}, \ldots, \frac{2p}{3}\}$, and a.
   Output: The x such that $g^x \equiv a \pmod{p}$

Summary of Where We Are

1. Finding primes p such that $p - 1 = 2q$, q a prime, EASY
2. Given such a p, finding generator g, EASY.
3. Given such a p, finding generator $g \in \{\frac{p}{3}, \ldots, \frac{2p}{3}\}$ EASY.
4. Given p, g, a finding $g^a \pmod{p}$ EASY.
5. The following problem thought to be hard:
   Input: prime p, generator $g \in \{\frac{p}{3}, \ldots, \frac{2p}{3}\}$, and a.
   Output: The x such that $g^x \equiv a \pmod{p}$

The problem thought to be hard is essentially the discrete log problem, though we have safeguarded against easy instances.

Summary of Where We Are

1. Finding primes p such that $p - 1 = 2q$, q a prime, EASY
2. Given such a p, finding generator g, EASY.
3. Given such a p, finding generator $g \in \{\frac{p}{3}, \ldots, \frac{2p}{3}\}$ EASY.
4. Given p, g, a finding $g^a \pmod{p}$ EASY.
5. The following problem thought to be hard:
   Input: prime p, generator $g \in \{\frac{p}{3}, \ldots, \frac{2p}{3}\}$, and a.
   Output: The x such that $g^x \equiv a \pmod{p}$

The problem thought to be hard is essentially the discrete log problem, though we have safeguarded against easy instances. We hope.

Convention (Possibly Repeated)

For the rest of the slides on **Diffie-Hellman Key Exchange** there will always be a prime p that we are considering and a generator $g \in \{\frac{p}{3}, \frac{2p}{3}\}$. We omit the bounds on g.

ALL arithmetic done from that point on is mod p.

ALL numbers are in $\{1, \ldots, p-1\}$.

The Diffie-Hellman Key Exchange

Alice & Bob want to establish a secret s w/o meeting.

The Diffie-Hellman Key Exchange

Alice & Bob want to establish a secret s w/o meeting.
Security parameter L.

The Diffie-Hellman Key Exchange

Alice & Bob want to establish a secret s w/o meeting.
Security parameter L.

1. Alice finds a (p, g), p of length L, g gen for $\mathbb{Z}_p^*$.

The Diffie-Hellman Key Exchange

Alice & Bob want to establish a secret s w/o meeting.
Security parameter L.

1. Alice finds a (p, g), p of length L, g gen for $\mathbb{Z}_p^*$.
2. Alice sends (p, g) to Bob (Eve can see it).

The Diffie-Hellman Key Exchange

Alice & Bob want to establish a secret s w/o meeting.
Security parameter L.

1. Alice finds a (p, g), p of length L, g gen for $\mathbb{Z}_p^*$.
2. Alice sends (p, g) to Bob (Eve can see it).
3. Alice picks rand a. Alice computes $g^a \pmod{p}$ and sends it to Bob (Eve can see it).

The Diffie-Hellman Key Exchange

Alice & Bob want to establish a secret s w/o meeting.
Security parameter L.

1. Alice finds a (p, g), p of length L, g gen for $\mathbb{Z}_p^*$.
2. Alice sends (p, g) to Bob (Eve can see it).
3. Alice picks rand a. Alice computes $g^a \pmod{p}$ and sends it to Bob (Eve can see it).
4. Bob picks rand b. Bob computes $g^b \pmod{p}$ and sends it to Alice (Eve can see it).

The Diffie-Hellman Key Exchange

Alice & Bob want to establish a secret s w/o meeting.
Security parameter L.

1. Alice finds a (p, g), p of length L, g gen for $\mathbb{Z}_p^*$.
2. Alice sends (p, g) to Bob (Eve can see it).
3. Alice picks rand a. Alice computes $g^a \pmod{p}$ and sends it to Bob (Eve can see it).
4. Bob picks rand b. Bob computes $g^b \pmod{p}$ and sends it to Alice (Eve can see it).
5. Alice computes $(g^b)^a = g^{ab} \pmod{p}$.

The Diffie-Hellman Key Exchange

Alice & Bob want to establish a secret s w/o meeting.
Security parameter L.

1. Alice finds a (p, g), p of length L, g gen for $\mathbb{Z}_p^*$.
2. Alice sends (p, g) to Bob (Eve can see it).
3. Alice picks rand a. Alice computes $g^a \pmod{p}$ and sends it to Bob (Eve can see it).
4. Bob picks rand b. Bob computes $g^b \pmod{p}$ and sends it to Alice (Eve can see it).
5. Alice computes $(g^b)^a = g^{ab} \pmod{p}$.
6. Bob computes $(g^a)^b = g^{ab} \pmod{p}$.

The Diffie-Hellman Key Exchange

Alice & Bob want to establish a secret s w/o meeting.
Security parameter L.

1. Alice finds a (p, g), p of length L, g gen for $\mathbb{Z}_p^*$.
2. Alice sends (p, g) to Bob (Eve can see it).
3. Alice picks rand a. Alice computes $g^a \pmod{p}$ and sends it to Bob (Eve can see it).
4. Bob picks rand b. Bob computes $g^b \pmod{p}$ and sends it to Alice (Eve can see it).
5. Alice computes $(g^b)^a = g^{ab} \pmod{p}$.
6. Bob computes $(g^a)^b = g^{ab} \pmod{p}$.
7. $s = g^{ab}$ is the shared secret.

The Diffie-Hellman Key Exchange

Alice & Bob want to establish a secret s w/o meeting.
Security parameter L.

1. Alice finds a (p, g), p of length L, g gen for $\mathbb{Z}_p^*$.
2. Alice sends (p, g) to Bob (Eve can see it).
3. Alice picks rand a. Alice computes $g^a \pmod{p}$ and sends it to Bob (Eve can see it).
4. Bob picks rand b. Bob computes $g^b \pmod{p}$ and sends it to Alice (Eve can see it).
5. Alice computes $(g^b)^a = g^{ab} \pmod{p}$.
6. Bob computes $(g^a)^b = g^{ab} \pmod{p}$.
7. $s = g^{ab}$ is the shared secret.

PRO: Alice and Bob can execute the protocol easily.

The Diffie-Hellman Key Exchange

Alice & Bob want to establish a secret s w/o meeting.
Security parameter L.

1. Alice finds a (p, g), p of length L, g gen for $\mathbb{Z}_p^*$.
2. Alice sends (p, g) to Bob (Eve can see it).
3. Alice picks rand a. Alice computes $g^a \pmod{p}$ and sends it to Bob (Eve can see it).
4. Bob picks rand b. Bob computes $g^b \pmod{p}$ and sends it to Alice (Eve can see it).
5. Alice computes $(g^b)^a = g^{ab} \pmod{p}$.
6. Bob computes $(g^a)^b = g^{ab} \pmod{p}$.
7. $s = g^{ab}$ is the shared secret.

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

The Diffie-Hellman Key Exchange

Alice & Bob want to establish a secret s w/o meeting.
Security parameter L.

1. Alice finds a (p, g), p of length L, g gen for $\mathbb{Z}_p^*$.
2. Alice sends (p, g) to Bob (Eve can see it).
3. Alice picks rand a. Alice computes $g^a \pmod{p}$ and sends it to Bob (Eve can see it).
4. Bob picks rand b. Bob computes $g^b \pmod{p}$ and sends it to Alice (Eve can see it).
5. Alice computes $(g^b)^a = g^{ab} \pmod{p}$.
6. Bob computes $(g^a)^b = g^{ab} \pmod{p}$.
7. $s = g^{ab}$ 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?

Have Students DO The DH Key Exchange

Pick out two students who I will call Alice and Bob.

Have Students DO The DH Key Exchange

Pick out two students who I will call Alice and Bob.

1. ALICE: Pick safe prime $256 \le p \le 511$ (so length 9).

Have Students DO The DH Key Exchange

Pick out two students who I will call Alice and Bob.

1. ALICE: Pick safe prime $256 \le p \le 511$ (so length 9).
2. ALICE: Find a generator g for $\mathbb{Z}_p^*$.

Have Students DO The DH Key Exchange

Pick out two students who I will call Alice and Bob.

1. ALICE: Pick safe prime $256 \le p \le 511$ (so length 9).
2. ALICE: Find a generator g for $\mathbb{Z}_p^*$.
3. ALICE: Yell out (p, g).

Have Students DO The DH Key Exchange

Pick out two students who I will call Alice and Bob.

1. ALICE: Pick safe prime $256 \le p \le 511$ (so length 9).
2. ALICE: Find a generator g for $\mathbb{Z}_p^*$.
3. ALICE: Yell out (p, g).
4. ALICE: Pick a rand $a \in \mathbb{Z}_p^*$.

Have Students DO The DH Key Exchange

Pick out two students who I will call Alice and Bob.

1. ALICE: Pick safe prime $256 \le p \le 511$ (so length 9).
2. ALICE: Find a generator g for $\mathbb{Z}_p^*$.
3. ALICE: Yell out (p, g).
4. ALICE: Pick a rand $a \in \mathbb{Z}_p^*$.
5. ALICE: Compute $g^a \pmod{p}$. YELL IT OUT.

Have Students DO The DH Key Exchange

Pick out two students who I will call Alice and Bob.

1. ALICE: Pick safe prime $256 \le p \le 511$ (so length 9).
2. ALICE: Find a generator g for $\mathbb{Z}_p^*$.
3. ALICE: Yell out (p, g).
4. ALICE: Pick a rand $a \in \mathbb{Z}_p^*$.
5. ALICE: Compute $g^a \pmod{p}$. YELL IT OUT.
6. BOB: Pick a rand $b \in \mathbb{Z}_p^*$.

Have Students DO The DH Key Exchange

Pick out two students who I will call Alice and Bob.

1. ALICE: Pick safe prime $256 \le p \le 511$ (so length 9).
2. ALICE: Find a generator g for $\mathbb{Z}_p^*$.
3. ALICE: Yell out (p, g).
4. ALICE: Pick a rand $a \in \mathbb{Z}_p^*$.
5. ALICE: Compute $g^a \pmod{p}$. YELL IT OUT.
6. BOB: Pick a rand $b \in \mathbb{Z}_p^*$.
7. BOB: Compute $g^b \pmod{p}$. YELL IT OUT.

Have Students DO The DH Key Exchange

Pick out two students who I will call Alice and Bob.

1. ALICE: Pick safe prime $256 \le p \le 511$ (so length 9).
2. ALICE: Find a generator g for $\mathbb{Z}_p^*$.
3. ALICE: Yell out (p, g).
4. ALICE: Pick a rand $a \in \mathbb{Z}_p^*$.
5. ALICE: Compute $g^a \pmod{p}$. YELL IT OUT.
6. BOB: Pick a rand $b \in \mathbb{Z}_p^*$.
7. BOB: Compute $g^b \pmod{p}$. YELL IT OUT.
8. ALICE: Compute $(g^b)^a \pmod{p}$.

Have Students DO The DH Key Exchange

Pick out two students who I will call Alice and Bob.

1. ALICE: Pick safe prime $256 \le p \le 511$ (so length 9).
2. ALICE: Find a generator g for $\mathbb{Z}_p^*$.
3. ALICE: Yell out (p, g).
4. ALICE: Pick a rand $a \in \mathbb{Z}_p^*$.
5. ALICE: Compute $g^a \pmod{p}$. YELL IT OUT.
6. BOB: Pick a rand $b \in \mathbb{Z}_p^*$.
7. BOB: Compute $g^b \pmod{p}$. YELL IT OUT.
8. ALICE: Compute $(g^b)^a \pmod{p}$.
9. BOB: Compute $(g^a)^b \pmod{p}$.

Have Students DO The DH Key Exchange

Pick out two students who I will call Alice and Bob.

1. ALICE: Pick safe prime $256 \le p \le 511$ (so length 9).
2. ALICE: Find a generator g for $\mathbb{Z}_p^*$.
3. ALICE: Yell out (p, g).
4. ALICE: Pick a rand $a \in \mathbb{Z}_p^*$.
5. ALICE: Compute $g^a \pmod{p}$. YELL IT OUT.
6. BOB: Pick a rand $b \in \mathbb{Z}_p^*$.
7. BOB: Compute $g^b \pmod{p}$. YELL IT OUT.
8. ALICE: Compute $(g^b)^a \pmod{p}$.
9. BOB: Compute $(g^a)^b \pmod{p}$.
10. At the count of 3 both yell out your number at the same time.

What Do We Really Know about Diffie-Hellman?

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

What Do We Really Know about Diffie-Hellman?

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

1. Eve sees g^a, g^b.

What Do We Really Know about Diffie-Hellman?

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.

What Do We Really Know about Diffie-Hellman?

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 $g^{ab} \pmod{p}$.

What Do We Really Know about Diffie-Hellman?

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 $g^{ab} \pmod{p}$.

Question: If Eve can crack DH then Eve can compute Discrete Log. **VOTE:** Y, N, UNKNOWN TO SCIENCE.

What Do We Really Know about Diffie-Hellman?

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 $g^{ab} \pmod{p}$.

Question: If Eve can crack DH then Eve can compute Discrete Log. **VOTE:** Y, N, UNKNOWN TO SCIENCE.

Unknown to Science

What Do We Really Know about Diffie-Hellman?

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 $g^{ab} \pmod{p}$.

Question: If Eve can crack DH then Eve can compute Discrete Log. **VOTE:** Y, N, UNKNOWN TO SCIENCE.

Unknown to Science

Question: If Eve can crack DH then Eve can compute ???.

Hardness Assumption

Definition Let *DHF* be the following function:
Inputs: p, g, g^a, g^b (note that a, b are not the input)

Outputs: g^{ab}.

Obvious Theorem: If Alice can crack Diffie-Hellman quickly then Alice can compute *DHF* quickly.

Hardness Assumption

Definition Let *DHF* be the following function:
Inputs: p, g, g^a, g^b (note that a, b are not the input)

Outputs: g^{ab}.

Obvious Theorem: If Alice can crack Diffie-Hellman quickly then Alice can compute *DHF* quickly.
Hardness assumption: *DHF* is hard to compute.

About the Hardness Assumption

Hardness assumption: DHF is hard to compute.

About the Hardness Assumption

Hardness assumption: DHF is hard to compute.
Do we believe the hardness assumption?

About the Hardness Assumption

Hardness assumption: DHF is hard to compute.

Do we believe the hardness assumption? Yes.

About the Hardness Assumption

Hardness assumption: *DHF* is hard to compute.
Do we believe the hardness assumption? Yes.

1. Nobody has found a way to solve DHF quickly that does not involve solving Discrete Log.

About the Hardness Assumption

Hardness assumption: *DHF* is hard to compute.
Do we believe the hardness assumption? Yes.

1. Nobody has found a way to solve DHF quickly that does not involve solving Discrete Log.
2. Discrete Log is believed to be hard.

About the Hardness Assumption

Hardness assumption: *DHF* is hard to compute.
Do we believe the hardness assumption? Yes.

1. Nobody has found a way to solve DHF quickly that does not involve solving Discrete Log.
2. Discrete Log is believed to be hard.
3. Still, would be nice to have a key exchange based on DL.

How Can Alice and Bob Use DH Key Exchange?

How Can Alice and Bob Use DH Key Exchange?

Alice and Bob want to set up a crypto system and NEVER MEET. Can DH help them?

How Can Alice and Bob Use DH Key Exchange?

Alice and Bob want to set up a crypto system and NEVER MEET. Can DH help them?

Example Alice wants to tell Bob to use Book Cipher with **Bounded Queries in Recursion theory by Gasarch and Martin**
`https://www.amazon.com/Bounded-Queries-Recursion-Progress-Computer/dp/1461268486`
a book that sold 2 copies last year AND Amazon has the author's name as **William Levine** (I do not now why).

How Can Alice and Bob Use DH Key Exchange?

Alice and Bob want to set up a crypto system and NEVER MEET. Can DH help them?

Example Alice wants to tell Bob to use Book Cipher with **Bounded Queries in Recursion theory by Gasarch and Martin**
`https://www.amazon.com/Bounded-Queries-Recursion-Progress-Computer/dp/1461268486`
a book that sold 2 copies last year AND Amazon has the author's name as **William Levine** (I do not now why).

How can Alice tell Bob this without meeting, possibly using DH? Discuss.

How Can Alice and Bob Use DH Key Exchange?

Alice and Bob want to set up a crypto system and NEVER MEET. Can DH help them?

Example Alice wants to tell Bob to use Book Cipher with **Bounded Queries in Recursion theory by Gasarch and Martin**
`https://www.amazon.com/Bounded-Queries-Recursion-Progress-Computer/dp/1461268486`
a book that sold 2 copies last year AND Amazon has the author's name as **William Levine** (I do not now why).

How can Alice tell Bob this without meeting, possibly using DH?
Discuss.
Next Slide continues this discussion.

With DH Alice and Bob do not the Message

Recall:

1. Alice finds a (p, g), p of length L, g gen for $\mathbb{Z}_p^*$.

With DH Alice and Bob do not the Message

Recall:

1. Alice finds a (p, g), p of length L, g gen for $\mathbb{Z}_p^*$.
2. Alice sends (p, g) to Bob (Eve can see it).

With DH Alice and Bob do not the Message

Recall:

1. Alice finds a (p, g), p of length L, g gen for $\mathbb{Z}_p^*$.
2. Alice sends (p, g) to Bob (Eve can see it).
3. Alice picks **rand** a. Alice computes g^a and broadcasts it.

With DH Alice and Bob do not the Message

Recall:

1. Alice finds a (p, g), p of length L, g gen for $\mathbb{Z}_p^*$.
2. Alice sends (p, g) to Bob (Eve can see it).
3. Alice picks **rand** a. Alice computes g^a and broadcasts it.
4. Bob picks **rand** b. Bob computes g^b and broadcasts it.

With DH Alice and Bob do not the Message

Recall:

1. Alice finds a (p, g), p of length L, g gen for $\mathbb{Z}_p^*$.
2. Alice sends (p, g) to Bob (Eve can see it).
3. Alice picks **rand** a. Alice computes g^a and broadcasts it.
4. Bob picks **rand** b. Bob computes g^b and broadcasts it.
5. Alice computes $(g^b)^a = g^{ab} \pmod{p}$.

With DH Alice and Bob do not the Message

Recall:

1. Alice finds a (p, g), p of length L, g gen for $\mathbb{Z}_p^*$.
2. Alice sends (p, g) to Bob (Eve can see it).
3. Alice picks **rand** a. Alice computes g^a and broadcasts it.
4. Bob picks **rand** b. Bob computes g^b and broadcasts it.
5. Alice computes $(g^b)^a = g^{ab} \pmod{p}$.
6. Bob computes $(g^a)^b = g^{ab} \pmod{p}$.

With DH Alice and Bob do not the Message

Recall:

1. Alice finds a (p, g), p of length L, g gen for $\mathbb{Z}_p^*$.
2. Alice sends (p, g) to Bob (Eve can see it).
3. Alice picks **rand** a. Alice computes g^a and broadcasts it.
4. Bob picks **rand** b. Bob computes g^b and broadcasts it.
5. Alice computes $(g^b)^a = g^{ab} \pmod{p}$.
6. Bob computes $(g^a)^b = g^{ab} \pmod{p}$.
7. $s = g^{ab}$ is the shared secret.

With DH Alice and Bob do not the Message

Recall:

1. Alice finds a (p, g), p of length L, g gen for $\mathbb{Z}_p^*$.
2. Alice sends (p, g) to Bob (Eve can see it).
3. Alice picks **rand** a. Alice computes g^a and broadcasts it.
4. Bob picks **rand** b. Bob computes g^b and broadcasts it.
5. Alice computes $(g^b)^a = g^{ab} \pmod{p}$.
6. Bob computes $(g^a)^b = g^{ab} \pmod{p}$.
7. $s = g^{ab}$ is the shared secret.

At the end Alice and Bob have $\boldsymbol{s}$

With DH Alice and Bob do not the Message

Recall:

1. Alice finds a (p, g), p of length L, g gen for $\mathbb{Z}_p^*$.
2. Alice sends (p, g) to Bob (Eve can see it).
3. Alice picks **rand** a. Alice computes g^a and broadcasts it.
4. Bob picks **rand** b. Bob computes g^b and broadcasts it.
5. Alice computes $(g^b)^a = g^{ab} \pmod{p}$.
6. Bob computes $(g^a)^b = g^{ab} \pmod{p}$.
7. $s = g^{ab}$ is the shared secret.

At the end Alice and Bob have s but **s has no meaning!**.

With DH Alice and Bob do not the Message

Recall:

1. Alice finds a (p, g), p of length L, g gen for $\mathbb{Z}_p^*$.
2. Alice sends (p, g) to Bob (Eve can see it).
3. Alice picks **rand** a. Alice computes g^a and broadcasts it.
4. Bob picks **rand** b. Bob computes g^b and broadcasts it.
5. Alice computes $(g^b)^a = g^{ab} \pmod{p}$.
6. Bob computes $(g^a)^b = g^{ab} \pmod{p}$.
7. $s = g^{ab}$ is the shared secret.

At the end Alice and Bob have s but **s has no meaning!**.
s is not going to be **Bounded Queries in Recursion Theory.**

With DH Alice and Bob do not the Message

Recall:

1. Alice finds a (p, g), p of length L, g gen for $\mathbb{Z}_p^*$.
2. Alice sends (p, g) to Bob (Eve can see it).
3. Alice picks **rand** a. Alice computes g^a and broadcasts it.
4. Bob picks **rand** b. Bob computes g^b and broadcasts it.
5. Alice computes $(g^b)^a = g^{ab} \pmod{p}$.
6. Bob computes $(g^a)^b = g^{ab} \pmod{p}$.
7. $s = g^{ab}$ is the shared secret.

At the end Alice and Bob have s but **s has no meaning!**.
s is not going to be **Bounded Queries in Recursion Theory.**
s is going to be some random number in $\{1, \ldots, p-1\}$.

How can Alice and Bob Use s?

s is random.

How can Alice and Bob Use s?

s is random. No meaning.

How can Alice and Bob Use s?

s is random. No meaning. Darn.

How can Alice and Bob Use s?

s is random. No meaning. Darn.

When life gives you a lemon, make lemonade.

How can Alice and Bob Use s?

s is random. No meaning. Darn.

When life gives you a lemon, make lemonade.

When life gives you a random string, use a one-time pad.

How can Alice and Bob Use s?

s is random. No meaning. Darn.

When life gives you a lemon, make lemonade.

When life gives you a random string, use a one-time pad.

1. Alice and Bob do DH and have shared string s.

How can Alice and Bob Use s?

s is random. No meaning. Darn.

When life gives you a lemon, make lemonade.

When life gives you a random string, use a one-time pad.

1. Alice and Bob do DH and have shared string s.
2. Alice uses s as the key for a 1-time pad to tell Bob the name of the Book for Book Cipher.

How can Alice and Bob Use s?

s is random. No meaning. Darn.

When life gives you a lemon, make lemonade.

When life gives you a random string, use a one-time pad.

1. Alice and Bob do DH and have shared string s.
2. Alice uses s as the key for a 1-time pad to tell Bob the name of the Book for Book Cipher.

This is not quite what people do but its the idea. Next slide is **El Gamal Public Key Crypto Systems** which is what people do.

Note really 1-Time Pad

Usual 1-Time Pad messages are bit strings. Use $\oplus$.

Note really 1-Time Pad

Usual 1-Time Pad messages are bit strings. Use $\oplus$.
In Next Protocol messages are elements of $\mathbb{Z}_p^*$. Use Mult Mod p.

ElGamal is DH Made Into an Enc System

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

We omit discussion of Hardness assumption (HW)

Misc Points about DH Key Exchange?

Possible Futures

Possible Futures

1. DL found to be easy, so DH is cracked.

Possible Futures

1. DL found to be easy, so DH is cracked.
2. DHF found to be easy, so DH is cracked.

Possible Futures

1. DL found to be easy, so DH is cracked.
2. DHF found to be easy, so DH is cracked.
3. Slightly better but still exp algorithms for DHF are found so Alice and Bob need to up their game, but DH still secure. (IMHO this is the most likely.)

Possible Futures

1. DL found to be easy, so DH is cracked.
2. DHF found to be easy, so DH is cracked.
3. Slightly better but still exp algorithms for DHF are found so Alice and Bob need to up their game, but DH still secure. (IMHO this is the most likely.)
4. DHF proven to be hard. (IMHO not gonna happen.)

Eve Could Think Outside The Box

Recall

Thm If Eve can crack DH quickly then Eve can compute DHF quickly.

Eve Could Think Outside The Box

Recall
Thm If Eve can crack DH quickly then Eve can compute DHF quickly.
So it seems as though we have a clean math problem that Eve has to solve to crack DH.

Eve Could Think Outside The Box

Recall
Thm If Eve can crack DH quickly then Eve can compute DHF quickly.
So it seems as though we have a clean math problem that Eve has to solve to crack DH.

1. Maginot Line is a good metaphor.

Eve Could Think Outside The Box

Recall

Thm If Eve can crack DH quickly then Eve can compute DHF quickly.

So it seems as though we have a clean math problem that Eve has to solve to crack DH.

1. Maginot Line is a good metaphor.
2. Eve could crack DH by putting on a Geek-Squad outfit and walking in to offer help.

Eve Could Think Outside The Box

Recall

Thm If Eve can crack DH quickly then Eve can compute DHF quickly.

So it seems as though we have a clean math problem that Eve has to solve to crack DH.

1. Maginot Line is a good metaphor.
2. Eve could crack DH by putting on a Geek-Squad outfit and walking in to offer help.
3. Eve could crack DH by bribing someone for a, b.

Eve Could Think Outside The Box

Recall
Thm If Eve can crack DH quickly then Eve can compute DHF quickly.
So it seems as though we have a clean math problem that Eve has to solve to crack DH.

1. Maginot Line is a good metaphor.
2. Eve could crack DH by putting on a Geek-Squad outfit and walking in to offer help.
3. Eve could crack DH by bribing someone for a, b.
4. Eve could measure how much time it takes for Bob to know the string and use that to narrow down the space of strings.

Eve in the Middle Attack

(Called **Man in the Middle Attack** in the literature.)
What if Eve could intercept both messages and replace them.

Eve in the Middle Attack

(Called **Man in the Middle Attack** in the literature.)
What if Eve could intercept both messages and replace them.

1. Alice sends g^a.

Eve in the Middle Attack

(Called **Man in the Middle Attack** in the literature.)
What if Eve could intercept both messages and replace them.

1. Alice sends g^a.
2. Eve intercepts the message, picks a random a', and instead sends on to Bob $g^{a'}$.

Eve in the Middle Attack

(Called **Man in the Middle Attack** in the literature.)
What if Eve could intercept both messages and replace them.

1. Alice sends g^a.
2. Eve intercepts the message, picks a random a', and instead sends on to Bob $g^{a'}$.
3. Eve lets Bob send g^b without interference.

Eve in the Middle Attack

(Called **Man in the Middle Attack** in the literature.)
What if Eve could intercept both messages and replace them.

1. Alice sends g^a.
2. Eve intercepts the message, picks a random a', and instead sends on to Bob $g^{a'}$.
3. Eve lets Bob send g^b without interference.
4. Alice thinks the shared secret string is g^{ab}.
   Bob thinks the shared secret string is $g^{a'b}$.

Eve in the Middle Attack

(Called **Man in the Middle Attack** in the literature.)
What if Eve could intercept both messages and replace them.

1. Alice sends g^a.
2. Eve intercepts the message, picks a random a', and instead sends on to Bob $g^{a'}$.
3. Eve lets Bob send g^b without interference.
4. Alice thinks the shared secret string is g^{ab}.
   Bob thinks the shared secret string is $g^{a'b}$.
   So Alice and Bob will not be able to communicate, which is a win for Eve.

Other Domains

Can do Diffie-Hellman with other structures that have these properties, that is, any Cyclic Group.

Other Domains

Can do Diffie-Hellman with other structures that have these properties, that is, any Cyclic Group.

In some cases this may be an advantage in that Eve's task is harder and Alice and Bob's task is not much harder.

Other Domains

Can do Diffie-Hellman with other structures that have these properties, that is, any Cyclic Group.

In some cases this may be an advantage in that Eve's task is harder and Alice and Bob's task is not much harder.

Example: Elliptic Curve Diffie-Hellman (actually used).

Other Domains

Can do Diffie-Hellman with other structures that have these properties, that is, any Cyclic Group.

In some cases this may be an advantage in that Eve's task is harder and Alice and Bob's task is not much harder.

Example: Elliptic Curve Diffie-Hellman (actually used).
Example: Braid Diffie-Hellman (not actually used).

A Serious Attack on Diffie-Helman! Or is it?

The paper **Imperfect Forward Secrecy: How Diffie-Helman Fails in Practice**

`https://weakdh.org/imperfect-forward-secrecy.pdf`

Claims the following:

A Serious Attack on Diffie-Helman! Or is it?

The paper **Imperfect Forward Secrecy: How Diffie-Helman Fails in Practice**

`https://weakdh.org/imperfect-forward-secrecy.pdf`

Claims the following:

1. 82% of all vulnerable servers use the same 512 sized group.

A Serious Attack on Diffie-Helman! Or is it?

The paper **Imperfect Forward Secrecy: How Diffie-Helman Fails in Practice**

`https://weakdh.org/imperfect-forward-secrecy.pdf`

Claims the following:

1. 82% of all vulnerable servers use the same 512 sized group.
2. After a week of preprocessing that group, they can crack DH on that group using an advanced DL algorithm.

A Serious Attack on Diffie-Helman! Or is it?

The paper **Imperfect Forward Secrecy: How Diffie-Helman Fails in Practice**

`https://weakdh.org/imperfect-forward-secrecy.pdf`

Claims the following:

1. 82% of all vulnerable servers use the same 512 sized group.
2. After a week of preprocessing that group, they can crack DH on that group using an advanced DL algorithm.
3. Their method can be adopted to larger groups.

A Serious Attack on Diffie-Helman! Or is it?

The paper **Imperfect Forward Secrecy: How Diffie-Helman Fails in Practice**

`https://weakdh.org/imperfect-forward-secrecy.pdf`

Claims the following:

1. 82% of all vulnerable servers use the same 512 sized group.
2. After a week of preprocessing that group, they can crack DH on that group using an advanced DL algorithm.
3. Their method can be adopted to larger groups.
4. For a 1024-sized group, they could not crack, but a nation with enough computing power could.

A Serious Attack on Diffie-Helman! Or is it?

The paper **Imperfect Forward Secrecy: How Diffie-Helman Fails in Practice**

`https://weakdh.org/imperfect-forward-secrecy.pdf`

Claims the following:

1. 82% of all vulnerable servers use the same 512 sized group.
2. After a week of preprocessing that group, they can crack DH on that group using an advanced DL algorithm.
3. Their method can be adopted to larger groups.
4. For a 1024-sized group, they could not crack, but a nation with enough computing power could.
5. The NSA may be using this approach.

A Serious Attack on Diffie-Helman! Or is it?

The paper **Imperfect Forward Secrecy: How Diffie-Helman Fails in Practice**

`https://weakdh.org/imperfect-forward-secrecy.pdf`

Claims the following:

1. 82% of all vulnerable servers use the same 512 sized group.
2. After a week of preprocessing that group, they can crack DH on that group using an advanced DL algorithm.
3. Their method can be adopted to larger groups.
4. For a 1024-sized group, they could not crack, but a nation with enough computing power could.
5. The NSA may be using this approach.

Sounds like DH is vulnerable! I posted about this on my blog and got responses (next slide).

A Serious Attack on Diffie-Helman! Or is it? (cont)

A Serious Attack on Diffie-Helman! Or is it? (cont)

1. **82% of all vulnerable servers use the same 512 sized group.** What about non-vulnerable servers? Not that many used this group when the paper came out.

A Serious Attack on Diffie-Helman! Or is it? (cont)

1. **82% of all vulnerable servers use the same 512 sized group.** What about non-vulnerable servers? Not that many used this group when the paper came out.
2. The paper **Critical Review of Imperfect Forward Secrecy** rebutted the attack. `https://www.cs.umd.edu/users/gasarch/COURSES/456/F20/lecpkprot/RSdh.pdf`

A Serious Attack on Diffie-Helman! Or is it? (cont)

1. **82% of all vulnerable servers use the same 512 sized group.** What about non-vulnerable servers? Not that many used this group when the paper came out.
2. The paper **Critical Review of Imperfect Forward Secrecy** rebutted the attack. `https://www.cs.umd.edu/users/gasarch/COURSES/456/F20/lecpkprot/RSdh.pdf`
3. Most systems no longer use the groups talked about in the paper.

A Serious Attack on Diffie-Helman! Or is it? (cont)

1. **82% of all vulnerable servers use the same 512 sized group.** What about non-vulnerable servers? Not that many used this group when the paper came out.
2. The paper **Critical Review of Imperfect Forward Secrecy** rebutted the attack.
`https://www.cs.umd.edu/users/gasarch/COURSES/456/F20/lecpkprot/RSdh.pdf`
3. Most systems no longer use the groups talked about in the paper.
4. NSA seems to be using a different attack.

A Serious Attack on Diffie-Helman! Or is it? (cont)

1. **82% of all vulnerable servers use the same 512 sized group.** What about non-vulnerable servers? Not that many used this group when the paper came out.
2. The paper **Critical Review of Imperfect Forward Secrecy** rebutted the attack.
   `https://www.cs.umd.edu/users/gasarch/COURSES/456/F20/lecpkprot/RSdh.pdf`
3. Most systems no longer use the groups talked about in the paper.
4. NSA seems to be using a different attack.
5. Jon Katz asked them for their code. They declined.

In My Humble Opinion

In My Humble Opinion

1. DH is safe from the attacks proposed in the paper.

In My Humble Opinion

1. DH is safe from the attacks proposed in the paper.
2. The paper still has an important message:

In My Humble Opinion

1. DH is safe from the attacks proposed in the paper.
2. The paper still has an important message:
   2.1 DO NOT use the same group (or the same p, g) all the time since some pre-computation may make it vulnerable.

In My Humble Opinion

1. DH is safe from the attacks proposed in the paper.
2. The paper still has an important message:
   2.1 DO NOT use the same group (or the same p, g) all the time since some pre-computation may make it vulnerable.
   2.2 UP your game! If L is so large that you think p of length L is safe. USE $10L$.

In My Humble Opinion

1. DH is safe from the attacks proposed in the paper.
2. The paper still has an important message:
   - 2.1 DO NOT use the same group (or the same p, g) all the time since some pre-computation may make it vulnerable.
   - 2.2 UP your game! If L is so large that you think p of length L is safe. USE $10L$.
   - 2.3 If you publish an **academic paper** about cracking DL, you should have the code and make it available. See next point.

In My Humble Opinion

1. DH is safe from the attacks proposed in the paper.
2. The paper still has an important message:
   2.1 DO NOT use the same group (or the same p, g) all the time since some pre-computation may make it vulnerable.
   2.2 UP your game! If L is so large that you think p of length L is safe. USE $10L$.
   2.3 If you publish an **academic paper** about cracking DL, you should have the code and make it available. See next point.
   2.4 If you actually worry about DH being cracked then tell the crypto companies or the government first. (See the fiction book **Factorman**. I reviewed it: `https://www.cs.umd.edu/users/gasarch/BLOGPAPERS/factorman.pdf`
)

BILL, STOP RECORDING LECTURE!!!!

BILL STOP RECORDING LECTURE!!!