Public Key Crypto: DH

September 25, 2019

The Diffie-Hellman 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-Hellman 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-Hellman 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-Hellman 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?

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 for $\mathbb{Z}_p^*$ that is not too big or small.

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 for $\mathbb{Z}_p^*$ that is not too big or small.
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 for $\mathbb{Z}_p^*$ that is not too big or small.
3. ALICE: Yell out (p, g).
4. ALICE: Pick a random $a \in \mathbb{Z}_p^*$ that is not too big or small. Write it down for later verification.

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 for $\mathbb{Z}_p^*$ that is not too big or small.
3. ALICE: Yell out (p, g).
4. ALICE: Pick a random $a \in \mathbb{Z}_p^*$ that is not too big or small. Write it down for later verification.
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 for $\mathbb{Z}_p^*$ that is not too big or small.
3. ALICE: Yell out (p, g).
4. ALICE: Pick a random $a \in \mathbb{Z}_p^*$ that is not too big or small. Write it down for later verification.
5. ALICE: Compute $g^a \pmod{p}$. YELL IT OUT.
6. BOB: Pick a random $b \in \mathbb{Z}_p^*$ that is not too big or small. Write it down for later verification.

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 for $\mathbb{Z}_p^*$ that is not too big or small.
3. ALICE: Yell out (p, g).
4. ALICE: Pick a random $a \in \mathbb{Z}_p^*$ that is not too big or small. Write it down for later verification.
5. ALICE: Compute $g^a \pmod{p}$. YELL IT OUT.
6. BOB: Pick a random $b \in \mathbb{Z}_p^*$ that is not too big or small. Write it down for later verification.
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 for $\mathbb{Z}_p^*$ that is not too big or small.
3. ALICE: Yell out (p, g).
4. ALICE: Pick a random $a \in \mathbb{Z}_p^*$ that is not too big or small. Write it down for later verification.
5. ALICE: Compute $g^a \pmod{p}$. YELL IT OUT.
6. BOB: Pick a random $b \in \mathbb{Z}_p^*$ that is not too big or small. Write it down for later verification.
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 for $\mathbb{Z}_p^*$ that is not too big or small.
3. ALICE: Yell out (p, g).
4. ALICE: Pick a random $a \in \mathbb{Z}_p^*$ that is not too big or small. Write it down for later verification.
5. ALICE: Compute $g^a \pmod{p}$. YELL IT OUT.
6. BOB: Pick a random $b \in \mathbb{Z}_p^*$ that is not too big or small. Write it down for later verification.
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 for $\mathbb{Z}_p^*$ that is not too big or small.
3. ALICE: Yell out (p, g).
4. ALICE: Pick a random $a \in \mathbb{Z}_p^*$ that is not too big or small. Write it down for later verification.
5. ALICE: Compute $g^a \pmod{p}$. YELL IT OUT.
6. BOB: Pick a random $b \in \mathbb{Z}_p^*$ that is not too big or small. Write it down for later verification.
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:

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

If Discrete Log Easy then DH is crackable

What about converse?

If DH is crackable then Discrete Log is Easy

VOTE: TRUE or FALSE or 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 $g^{ab} \pmod{p}$.

If Discrete Log Easy then DH is crackable

What about converse?

If DH is crackable then Discrete Log is Easy

VOTE: TRUE or FALSE or UNKNOWN TO SCIENCE
UNKNOWN TO SCIENCE.
Note: In ugrad math classes rare to have a statement that is UNKNOWN TO SCIENCE. Discuss.

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 (HA): f is hard to compute.

One can show, assuming the hardness assumption, that DH is hard to crack.

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 (HA): f is hard to compute.

One can show, assuming the hardness assumption, that DH is hard to crack.

But any such proof assumes Eve has limits.

Next slide gives example.

What Do Proofs of Security Assume?

Silly Example A proof that Eve cannot find out the secret assumes that Eve cannot bribe Alice into revealing the secret.

What Do Proofs of Security Assume?

Silly Example A proof that Eve cannot find out the secret assumes that Eve cannot bribe Alice into revealing the secret.

Serious Example Timing Attacks. There have been successful attacks that measure how much time it takes Bob to compute g^b to cut down the search space. For example: OH, Bob took a short time, maybe b in binary does not have that many 1's in it.

What Do Proofs of Security Assume?

Silly Example A proof that Eve cannot find out the secret assumes that Eve cannot bribe Alice into revealing the secret.

Serious Example Timing Attacks. There have been successful attacks that measure how much time it takes Bob to compute g^b to cut down the search space. For example: OH, Bob took a short time, maybe b in binary does not have that many 1's in it.

Upshot We will not be getting into proofs by security. However, be forewarned that any proof of security should be viewed as a way to differentiate what attacks won't work and what attacks will.

What Do Proofs of Security Assume?

Silly Example A proof that Eve cannot find out the secret assumes that Eve cannot bribe Alice into revealing the secret.

Serious Example Timing Attacks. There have been successful attacks that measure how much time it takes Bob to compute g^b to cut down the search space. For example: OH, Bob took a short time, maybe b in binary does not have that many 1's in it.

Upshot We will not be getting into proofs by security. However, be forewarned that any proof of security should be viewed as a way to differentiate what attacks won't work and what attacks will.

Reading Look up the Maginot Line.

What Could be True?

(Recall that HA is the Hardness Assumption.)
The following are all possible:

What Could be True?

(Recall that HA is the Hardness Assumption.)
The following are all possible:

1) DL is easy. Then DH is crackable.

What Could be True?

(Recall that HA is the Hardness Assumption.)
The following are all possible:

1) DL is easy. Then DH is crackable.

2) DL is hard, HA is false. DH is crackable, though DL is hard!!

What Could be True?

(Recall that HA is the Hardness Assumption.)
The following are all possible:

1) DL is easy. Then DH is crackable.

2) DL is hard, HA is false. DH is crackable, though DL is hard!!

3) DL is hard, HA is true, but DH is crackable by other means.
Timing Attacks. Must rethink our model of security.

What Could be True?

(Recall that HA is the Hardness Assumption.)
The following are all possible:

1) DL is easy. Then DH is crackable.

2) DL is hard, HA is false. DH is crackable, though DL is hard!!

3) DL is hard, HA is true, but DH is crackable by other means. Timing Attacks. Must rethink our model of security.

4) DL is hard, HA 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?

What About $\mathbb{Z}_p^*$ Did Diffie-Hellman Use?

1. Multiplication. We DID NOT use addition. So we used $\mathbb{Z}_p^*$.
2. $\mathbb{Z}_p^*$ has a generator.
3. g^a is easy to compute.
4. Discrete Log is (though to be) hard to compute.
5. (g^a, g^b) to g^{ab} is (thought to be) hard to compute.

What About $\mathbb{Z}_p^*$ Did Diffie-Hellman Use?

1. Multiplication. We DID NOT use addition. So we used $\mathbb{Z}_p^*$.
2. $\mathbb{Z}_p^*$ has a generator.
3. g^a is easy to compute.
4. Discrete Log is (though to be) hard to compute.
5. (g^a, g^b) to g^{ab} is (thought to be) hard to compute.

Can do DH over any cyclic group with these properties. 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.

What About $\mathbb{Z}_p^*$ Did Diffie-Hellman Use?

1. Multiplication. We DID NOT use addition. So we used $\mathbb{Z}_p^*$.
2. $\mathbb{Z}_p^*$ has a generator.
3. g^a is easy to compute.
4. Discrete Log is (though to be) hard to compute.
5. (g^a, g^b) to g^{ab} is (thought to be) hard to compute.

Can do DH over any cyclic group with these properties. 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 Succesful Attacks on DH . . . Maybe

A Succesful Attacks on DH . . . Maybe

Paper: Imperfect Forward Secrecy: How DH Fails in Practice.
(Paper on Course Website.)
Claims to Breaks DH Uses the following

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

A Succesful Attacks on DH . . . Maybe

Paper: Imperfect Forward Secrecy: How DH Fails in Practice.
(Paper on Course Website.)
Claims to Breaks DH Uses the following

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

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

A Succesful Attacks on DH . . . Maybe

Paper: Imperfect Forward Secrecy: How DH Fails in Practice. (Paper on Course Website.)
Claims to Breaks DH Uses the following

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

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

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

A Succesful Attacks on DH . . . Maybe

Paper: Imperfect Forward Secrecy: How DH Fails in Practice.
(Paper on Course Website.)
Claims to Breaks DH Uses the following

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

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 DL is a bit easier (later).

A Succesful Attacks on DH . . . Maybe

Paper: Imperfect Forward Secrecy: How DH Fails in Practice.
(Paper on Course Website.)
Claims to Breaks DH Uses the following

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

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 DL is a bit easier (later).
After publishing the paper. . .

A Succesful Attacks on DH . . . Maybe

Paper: Imperfect Forward Secrecy: How DH Fails in Practice. (Paper on Course Website.)
Claims to Breaks DH Uses the following

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

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 DL is a bit easier (later).
After publishing the paper. . . The authors have not been heard from since!

A Succesful Attacks on DH . . . Maybe

Paper: Imperfect Forward Secrecy: How DH Fails in Practice. (Paper on Course Website.)
Claims to Breaks DH Uses the following

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

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 DL is a bit easier (later).
After publishing the paper. . . The authors have not been heard from since!
Just Kidding.

I Blogged Asking What is Known about This Approach

Pointer to my blog entry is on course website.

1. The approach needs a lot of precomputation. If people change keys often enough, that thwarts the attack.

I Blogged Asking What is Known about This Approach

Pointer to my blog entry is on course website.

1. The approach needs a lot of precomputation. If people change keys often enough, that thwarts the attack.
2. There has been another paper that challenged the claims.

I Blogged Asking What is Known about This Approach

Pointer to my blog entry is on course website.

1. The approach needs a lot of precomputation. If people change keys often enough, that thwarts the attack.
2. There has been another paper that challenged the claims.
3. By the time the paper came out many people had already switched to Elliptic Curve Crypto.

I Blogged Asking What is Known about This Approach

Pointer to my blog entry is on course website.

1. The approach needs a lot of precomputation. If people change keys often enough, that thwarts the attack.
2. There has been another paper that challenged the claims.
3. By the time the paper came out many people had already switched to Elliptic Curve Crypto.
4. When asked for their code, the authors did not supply it.

My Opinion

1. Paper was published in Academic Journal, hence posting code is expected. This is the big negative.
2. I suspect that the authors had a byte of bad timing—as they were writing the paper people upped their game— larger parameters, different settings.
3. Their paper gives us things to watch out for, so I respect that.
4. Some of the comments on my blog, and emails I got were nasty to the authors. Thats unfair.

Variants of Standard Diffie-Helman

Recall the Diffie-Helman Key Exchange

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

Why does Alice: rand $a \in \{\frac{p}{3}, \ldots, \frac{2p}{3}\}$.
Why not $a \in \{1, \ldots, p-1\}$? Discuss

Recall the Diffie-Helman Key Exchange

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

Why does Alice: rand $a \in \{\frac{p}{3}, \ldots, \frac{2p}{3}\}$.
Why not $a \in \{1, \ldots, p-1\}$? Discuss
If g is small and a is small then Eve can determine a from g^a.

Recall the Diffie-Helman Key Exchange

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

Why does Alice: rand $a \in \{\frac{p}{3}, \ldots, \frac{2p}{3}\}$.
Why not $a \in \{1, \ldots, p-1\}$? Discuss
If g is small and a is small then Eve can determine a from g^a.
But: Eve can compute $g^1, \ldots, g^L$ and if she sees any of those she knows.

Example

$p = 1013$
$g = 5$
$a = 6$
Eve computes ahead of time:
$5^0 = 1$
$5^1 = 5$
$5^2 = 25$
$5^3 = 125$
$5^4 = 625$
$5^5 = 86$
$5^6 = 430$
If Eve sees Alice 430 then she knows $a = 6$
Nothing special about a being small.

Example

$p = 1013$
$g = 40$
$a \in \{\frac{p}{3}, \ldots, \frac{2p}{3}\} = \{337, \ldots, 674\}$
Note: We assume that Eve KNOWS these endpoints.
Eve computes
$40^{337} \equiv 919$
$40^{338} \equiv 292$
$40^{339} \equiv 537$
$40^{340} \equiv 207$
$40^{341} \equiv 176$
$40^{342} \equiv 962$
$40^{343} \equiv 999$
If Eve sees Alice send any of $919, 292, 537, 207, 176, 962, 999$ then she knows a
g was big, a was big. Didn't help!

Example

$p = 1013$
$g = 40$
$a \in \{\frac{p}{3}, \ldots, \frac{2p}{3}\} = \{337, \ldots, 674\}$
Note: We assume that Eve KNOWS these endpoints.
Eve computes
$40^{337} \equiv 919$
$40^{338} \equiv 292$
$40^{339} \equiv 537$
$40^{340} \equiv 207$
$40^{341} \equiv 176$
$40^{342} \equiv 962$
$40^{343} \equiv 999$
If Eve sees Alice send any of $919, 292, 537, 207, 176, 962, 999$ then she knows a
g was big, a was big. Didn't help!
Of course, Eve has to get VERY LUCKY.

Diffie-Helman as Often Practiced

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: rand $a \in \{1, \ldots, p-1\}$, sends g^a.
4. Bob: rand $b \in \{1, \ldots, p-1\}$, sends g^b.
5. Alice:$(g^b)^a = g^{ab}$. Bob:$(g^a)^b = g^{ab}$. g^{ab} is shared secret.

Eve comp $g^1, \ldots, g^L$. If $a \in \{1, \ldots, L\}$ Eve knows a.
Debatable Not really a problem:
Either

1. If L is small then Eve would have to get LUCKY to find a.
2. If L is large then Eve is doing LOTS OF computation.

Upshot: a, g small did not make attack much easier for Eve.

Is There Harm In Restricting a, b?

Does requiring $a, b \in \{\frac{p}{3}, \ldots, \frac{2p}{3}\}$ help?

Is There Harm In Restricting a, b?

Does requiring $a, b \in \{\frac{p}{3}, \ldots, \frac{2p}{3}\}$ help?

- Yes: Some obvious easy cases of DL are avoided.
- No: Eve can pre-compute any small number of cases anyway.

Is There Harm In Restricting a, b?

Does requiring $a, b \in \{\frac{p}{3}, \ldots, \frac{2p}{3}\}$ help?

- Yes: Some obvious easy cases of DL are avoided.
- No: Eve can pre-compute any small number of cases anyway.

Does requiring $a, b \in \{\frac{p}{3}, \ldots, \frac{2p}{3}\}$ hurt?

Is There Harm In Restricting a, b?

Does requiring $a, b \in \{\frac{p}{3}, \ldots, \frac{2p}{3}\}$ help?

- Yes: Some obvious easy cases of DL are avoided.
- No: Eve can pre-compute any small number of cases anyway.

Does requiring $a, b \in \{\frac{p}{3}, \ldots, \frac{2p}{3}\}$ hurt?
Key space is smaller, making it easier for Eve.

Is There Harm In Restricting a, b?

Does requiring $a, b \in \{\frac{p}{3}, \ldots, \frac{2p}{3}\}$ help?

- Yes: Some obvious easy cases of DL are avoided.
- No: Eve can pre-compute any small number of cases anyway.

Does requiring $a, b \in \{\frac{p}{3}, \ldots, \frac{2p}{3}\}$ hurt?
Key space is smaller, making it easier for Eve.
A matter of opinion. I think it helps. Others disagree.

How Useful is Diffie-Hellman

CON: Alice and Bob share g^{ab} which is not in their control.

CAVEAT: DH is not a cipher.

PRO: Alice and Bob can use g^{ab} to transmit a key for a cipher.
CON: Alice and Bob share g^{ab} which is not in their control.
Discuss

How Useful is Diffie-Hellman

CON: Alice and Bob share g^{ab} which is not in their control.

CAVEAT: DH is not a cipher.

PRO: Alice and Bob can use g^{ab} to transmit a key for a cipher.
CON: Alice and Bob share g^{ab} which is not in their control.
Discuss
Alice and Bob do not control the key. Is that bad?

Using Diffie-Hellman to Transmit a Key

Using Diffie-Hellman to Transmit a Key

Shift Cipher: Can use DH to transmit a key that is your shift. You don't get to choose the shift. Thats fine—the shift was chosen at random anyway.

Using Diffie-Hellman to Transmit a Key

Shift Cipher: Can use DH to transmit a key that is your shift. You don't get to choose the shift. Thats fine—the shift was chosen at random anyway.

Affine, Matrix, Vig: Similar.

Using Diffie-Hellman to Transmit a Key

Shift Cipher: Can use DH to transmit a key that is your shift. You don't get to choose the shift. Thats fine—the shift was chosen at random anyway.

Affine, Matrix, Vig: Similar.

One Time Pad: My favorite. DH gives Alice and Bob a Random secret key. So this is perfect!

Using Diffie-Hellman to Transmit a Key

Shift Cipher: Can use DH to transmit a key that is your shift. You don't get to choose the shift. Thats fine—the shift was chosen at random anyway.

Affine, Matrix, Vig: Similar.

One Time Pad: My favorite. DH gives Alice and Bob a Random secret key. So this is perfect!

How Really Used: DH is often used to transmit the parameters of a random number generator, and that is used for a Faux-one-time-pad.

Recall Diffie-Hellman

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 Let's 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 Made Into an Enc System

1. Alice and Bob do Diffie Hellman.
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)

ElGasarch is DH Made into an Enc System

1. Alice and Bob do Diffie Hellman over mod p. Let $n = \lceil \lg p \rceil$. All elements of $\mathbb{Z}_p^*$ are n-bit strings.
2. Alice and Bob share secret $s = g^{ab}$. View as a bit string.
3. To send m, Alice sends $c = m \oplus s$ (this is NOT mod p)
4. To decrypt, Bob computes $c \oplus s = m \oplus s \oplus s = mp$ (this is NOT mod p)

Why is ElGamal used and ElGasarch is not? Discuss

ElGasarch is DH Made into an Enc System

1. Alice and Bob do Diffie Hellman over mod p. Let $n = \lceil \lg p \rceil$. All elements of $\mathbb{Z}_p^*$ are n-bit strings.
2. Alice and Bob share secret $s = g^{ab}$. View as a bit string.
3. To send m, Alice sends $c = m \oplus s$ (this is NOT mod p)
4. To decrypt, Bob computes $c \oplus s = m \oplus s \oplus s = mp$ (this is NOT mod p)

Why is ElGamal used and ElGasarch is not? Discuss

Example: $p = 23$. The elements are $\{0, \ldots, 22\}$. $0, \ldots, 15$ use 4 bits. $16, \ldots, 22$ use 5 bits. So if all use 5 bits then $15/22 \sim 0.68$ of the strings have a 0 as first bit. Not Random Enough.

Could ElGasarch work with some variant of DH? Discuss

ElGasarch is DH Made into an Enc System

1. Alice and Bob do Diffie Hellman over mod p. Let $n = \lceil \lg p \rceil$. All elements of $\mathbb{Z}_p^*$ are n-bit strings.
2. Alice and Bob share secret $s = g^{ab}$. View as a bit string.
3. To send m, Alice sends $c = m \oplus s$ (this is NOT mod p)
4. To decrypt, Bob computes $c \oplus s = m \oplus s \oplus s = mp$ (this is NOT mod p)

Why is ElGamal used and ElGasarch is not? Discuss

Example: $p = 23$. The elements are $\{0, \ldots, 22\}$. $0, \ldots, 15$ use 4 bits. $16, \ldots, 22$ use 5 bits. So if all use 5 bits then $15/22 \sim 0.68$ of the strings have a 0 as first bit. Not Random Enough.

Could ElGasarch work with some variant of DH? Discuss

Would need to do DH over a group (1) with power-of-2 elts, (2) DL is hard, (3) mult is easy. Do any exist? Do not know.