Disc 2d

Andy

UC Berkeley

June 28, 2018

Mini Lecture: One Time Pad

1. Unbreakable encryption method

Mini Lecture: One Time Pad

1. Unbreakable encryption method
2. "a" + "a" = "b", "a" + "c" = "d"

Mini Lecture: One Time Pad

1. Unbreakable encryption method
2. "a" + "a" = "b", "a" + "c" = "d"
3. Alice and Bob agree on a random cipher as long as the message

Mini Lecture: One Time Pad

1. Unbreakable encryption method
2. "a" + "a" = "b", "a" + "c" = "d"
3. Alice and Bob agree on a random cipher as long as the message
4. "addekficjladkfjghe"

Mini Lecture: One Time Pad

1. Unbreakable encryption method
2. "a" + "a" = "b", "a" + "c" = "d"
3. Alice and Bob agree on a random cipher as long as the message
4. "addekficjladkfjghe"
5. "Plain Text" + cipher = encrypted text

Mini Lecture: One Time Pad

1. Unbreakable encryption method
2. "a" + "a" = "b", "a" + "c" = "d"
3. Alice and Bob agree on a random cipher as long as the message
4. "addekficjladkfjghe"
5. "Plain Text" + cipher = encrypted text
6. "abc" what does this decrypt to?

Mini Lecture: One Time Pad

1. Unbreakable encryption method
2. "a" + "a" = "b", "a" + "c" = "d"
3. Alice and Bob agree on a random cipher as long as the message
4. "addekficjladkfjghe"
5. "Plain Text" + cipher = encrypted text
6. "abc" what does this decrypt to?
7. "dog", "cat", "bob", "lol",.... any 3 letter word...

Mini Lecture: One Time Pad

1. Unbreakable encryption method
2. "a" + "a" = "b", "a" + "c" = "d"
3. Alice and Bob agree on a random cipher as long as the message
4. "addekficjladkfjghe"
5. "Plain Text" + cipher = encrypted text
6. "abc" what does this decrypt to?
7. "dog", "cat", "bob", "lol",.... any 3 letter word...
8. Discovered by a Civil War Vet / Stanford Trustee, AT+T, some Russians, ... etc.

Mini Lecture: RSA

1. James Ellis, Malcolm Williamson of GCHQ discover Diffie-Hellman key exchange, 1969

Mini Lecture: RSA

1. James Ellis, Malcolm Williamson of GCHQ discover Diffie-Hellman key exchange, 1969
2. This method allows 2 parties to agree on some number n without revealing it

Mini Lecture: RSA

1. James Ellis, Malcolm Williamson of GCHQ discover Diffie-Hellman key exchange, 1969
2. This method allows 2 parties to agree on some number n without revealing it
3. Clifford Cocks, of GCHQ, discovers RSA, 1973

Mini Lecture: RSA

1. James Ellis, Malcolm Williamson of GCHQ discover Diffie-Hellman key exchange, 1969
2. This method allows 2 parties to agree on some number n without revealing it
3. Clifford Cocks, of GCHQ, discovers RSA, 1973
4. Diffie and Hellman rediscover DH key exchange, 1976

Mini Lecture: RSA

1. James Ellis, Malcolm Williamson of GCHQ discover Diffie-Hellman key exchange, 1969
2. This method allows 2 parties to agree on some number n without revealing it
3. Clifford Cocks, of GCHQ, discovers RSA, 1973
4. Diffie and Hellman rediscover DH key exchange, 1976
5. Diffie and Hellman can't figure out a good 1 way function.

Mini Lecture: RSA

1. James Ellis, Malcolm Williamson of GCHQ discover Diffie-Hellman key exchange, 1969
2. This method allows 2 parties to agree on some number n without revealing it
3. Clifford Cocks, of GCHQ, discovers RSA, 1973
4. Diffie and Hellman rediscover DH key exchange, 1976
5. Diffie and Hellman can't figure out a good 1 way function.
6. Ron Rivest, Adi Shamir, and Leonard Adleman rediscover RSA, 1977

Mini Lecture: RSA

1. James Ellis, Malcolm Williamson of GCHQ discover Diffie-Hellman key exchange, 1969
2. This method allows 2 parties to agree on some number n without revealing it
3. Clifford Cocks, of GCHQ, discovers RSA, 1973
4. Diffie and Hellman rediscover DH key exchange, 1976
5. Diffie and Hellman can't figure out a good 1 way function.
6. Ron Rivest, Adi Shamir, and Leonard Adleman rediscover RSA, 1977
7. Rivest comes up with a good one way function after getting drunk at Passover

Mini Lecture: RSA

1. James Ellis, Malcolm Williamson of GCHQ discover Diffie-Hellman key exchange, 1969
2. This method allows 2 parties to agree on some number n without revealing it
3. Clifford Cocks, of GCHQ, discovers RSA, 1973
4. Diffie and Hellman rediscover DH key exchange, 1976
5. Diffie and Hellman can't figure out a good 1 way function.
6. Ron Rivest, Adi Shamir, and Leonard Adleman rediscover RSA, 1977
7. Rivest comes up with a good one way function after getting drunk at Passover
8. Those GCHQ guys eventually get credit for their work in 1997.

Mini Lecture: RSA

1. James Ellis, Malcolm Williamson of GCHQ discover Diffie-Hellman key exchange, 1969
2. This method allows 2 parties to agree on some number n without revealing it
3. Clifford Cocks, of GCHQ, discovers RSA, 1973
4. Diffie and Hellman rediscover DH key exchange, 1976
5. Diffie and Hellman can't figure out a good 1 way function.
6. Ron Rivest, Adi Shamir, and Leonard Adleman rediscover RSA, 1977
7. Rivest comes up with a good one way function after getting drunk at Passover
8. Those GCHQ guys eventually get credit for their work in 1997.
9. RSA is the most copied algorithm in the world!

Mini Lecture: RSA

1. Analogy:

Mini Lecture: RSA

1. Analogy:
2. Combination locks: Anyone can lock them, only the owner can unlock them

Mini Lecture: RSA

1. Analogy:
2. Combination locks: Anyone can lock them, only the owner can unlock them
3. You want to send Andy a message. You ask Andy for his combo lock and a box

Mini Lecture: RSA

1. Analogy:
2. Combination locks: Anyone can lock them, only the owner can unlock them
3. You want to send Andy a message. You ask Andy for his combo lock and a box
4. You stuff your message in the box, and slam the lock shut

Mini Lecture: RSA

1. Analogy:
2. Combination locks: Anyone can lock them, only the owner can unlock them
3. You want to send Andy a message. You ask Andy for his combo lock and a box
4. You stuff your message in the box, and slam the lock shut
5. Only Andy knows the combination, so only he can open it.

Mini Lecture: RSA Scheme

1. m is message, N, e are 2 public keys, d is secret private key

Mini Lecture: RSA Scheme

1. m is message, N, e are 2 public keys, d is secret private key
2. Bob wants to send Alice a secret message

Mini Lecture: RSA Scheme

1. m is message, N, e are 2 public keys, d is secret private key
2. Bob wants to send Alice a secret message
3. Alice will generate N, e, d, and share N, e with Bob.

Mini Lecture: RSA Scheme

1. m is message, N, e are 2 public keys, d is secret private key
2. Bob wants to send Alice a secret message
3. Alice will generate N, e, d, and share N, e with Bob.
4. Bob will calculate $m^e \bmod N$ and send the result to Alice

Mini Lecture: RSA Scheme

1. m is message, N, e are 2 public keys, d is secret private key
2. Bob wants to send Alice a secret message
3. Alice will generate N, e, d, and share N, e with Bob.
4. Bob will calculate $m^e \bmod N$ and send the result to Alice
5. Call the encrypted message c, for ciphertext

Mini Lecture: RSA Scheme

1. m is message, N, e are 2 public keys, d is secret private key
2. Bob wants to send Alice a secret message
3. Alice will generate N, e, d, and share N, e with Bob.
4. Bob will calculate m^e mod N and send the result to Alice
5. Call the encrypted message c, for ciphertext
6. Alice will do $c^d \equiv m^{e^d} \equiv m^{ed} \equiv m \bmod N$.

Mini Lecture: Totient

1. $\phi(N) =$ number of positive integers relatively prime to n, up to n.

Mini Lecture: Totient

1. $\phi(N) =$ number of positive integers relatively prime to n, up to n.
2. $\phi(9) = 6$, since $1, 2, 4, 5, 7, 8$

Mini Lecture: Totient

1. $\phi(N) =$ number of positive integers relatively prime to n, up to n.
2. $\phi(9) = 6$, since $1, 2, 4, 5, 7, 8$
3. Totient is hard to calculate... unless you know the prime factors!

Mini Lecture: Totient

1. $\phi(N) =$ number of positive integers relatively prime to n, up to n.
2. $\phi(9) = 6$, since $1, 2, 4, 5, 7, 8$
3. Totient is hard to calculate... unless you know the prime factors!
4. $\phi(N) = (p-1)(q-1)(r-1)(s-1)...$ where $p, q, r, s, ...$ are prime factors

Mini Lecture: RSA Generation

1. Alice will generate N, e, d, and share N, e with Bob.

Mini Lecture: RSA Generation

1. Alice will generate N, e, d, and share N, e with Bob.
2. $p, q =$ large prime numbers

Mini Lecture: RSA Generation

1. Alice will generate N, e, d, and share N, e with Bob.
2. $p, q =$ large prime numbers
3. $N = pq$

Mini Lecture: RSA Generation

1. Alice will generate N, e, d, and share N, e with Bob.
2. $p, q =$ large prime numbers
3. $N = pq$
4. Calclate $\phi(N) = (p-1)(q-1)$

Mini Lecture: RSA Generation

1. Alice will generate N, e, d, and share N, e with Bob.
2. $p, q =$ large prime numbers
3. $N = pq$
4. Calclate $\phi(N) = (p-1)(q-1)$
5. Choose e st it is relatively prime to $\phi(N)$

Mini Lecture: RSA Generation

1. Alice will generate N, e, d, and share N, e with Bob.
2. $p, q =$ large prime numbers
3. $N = pq$
4. Calclate $\phi(N) = (p-1)(q-1)$
5. Choose e st it is relatively prime to $\phi(N)$
6. $d \equiv e^{-1} \bmod \phi(N)$

Mini Lecture: RSA Generation

1. Alice will generate N, e, d, and share N, e with Bob.
2. $p, q =$ large prime numbers
3. $N = pq$
4. Calclate $\phi(N) = (p-1)(q-1)$
5. Choose e st it is relatively prime to $\phi(N)$
6. $d \equiv e^{-1} \bmod \phi(N)$
7. RSA is secure!!!!!!

Mini Lecture: RSA Generation

1. Alice will generate N, e, d, and share N, e with Bob.
2. $p, q =$ large prime numbers
3. $N = pq$
4. Calclate $\phi(N) = (p-1)(q-1)$
5. Choose e st it is relatively prime to $\phi(N)$
6. $d \equiv e^{-1} \bmod \phi(N)$
7. RSA is secure!!!!!!
8. Lets try to break it. We try to solve for d. Guess and check? ✗

Mini Lecture: RSA Generation

1. Alice will generate N, e, d, and share N, e with Bob.
2. $p, q =$ large prime numbers
3. $N = pq$
4. Calclate $\phi(N) = (p-1)(q-1)$
5. Choose e st it is relatively prime to $\phi(N)$
6. $d \equiv e^{-1} \bmod \phi(N)$
7. RSA is secure!!!!!!
8. Lets try to break it. We try to solve for d. Guess and check? ✗
9. We have e. We need $\phi(N)$, What is ϕ. Guess and check? ✗

Mini Lecture: RSA Generation

1. Alice will generate N, e, d, and share N, e with Bob.
2. $p, q =$ large prime numbers
3. $N = pq$
4. Calclate $\phi(N) = (p-1)(q-1)$
5. Choose e st it is relatively prime to $\phi(N)$
6. $d \equiv e^{-1} \bmod \phi(N)$
7. RSA is secure!!!!!!
8. Lets try to break it. We try to solve for d. Guess and check? ✗
9. We have e. We need $\phi(N)$, What is ϕ. Guess and check? ✗
10. If we could factor N into p, q, we can get ϕ.

Mini Lecture: RSA Generation

1. Alice will generate N, e, d, and share N, e with Bob.
2. $p, q =$ large prime numbers
3. $N = pq$
4. Calclate $\phi(N) = (p-1)(q-1)$
5. Choose e st it is relatively prime to $\phi(N)$
6. $d \equiv e^{-1} \mod \phi(N)$
7. RSA is secure!!!!!!
8. Lets try to break it. We try to solve for d. Guess and check? ✗
9. We have e. We need $\phi(N)$, What is ϕ. Guess and check? ✗
10. If we could factor N into p, q, we can get ϕ.
11. Factoring is guessing and checking... ✗

Q3

1. $d = e^{-1} \bmod (p-1)(q-1)(r-1)$.

Q3

1. $d = e^{-1} \bmod (p-1)(q-1)(r-1)$.
2. $x^{ed} - x \equiv 0 mod N$

Q3

1. $d = e^{-1} \bmod (p-1)(q-1)(r-1)$.
2. $x^{ed} - x \equiv 0 mod N$
3. $x(x^{ed-1} - 1)$ must be divisible by p, q, r

Q3

1. $d = e^{-1} \bmod (p-1)(q-1)(r-1)$.
2. $x^{ed} - x \equiv 0 mod N$
3. $x(x^{ed-1} - 1)$ must be divisible by p, q, r
4. $x(x^{k(p-1)(q-1)(r-1)+1-1} - 1)$ must be divisible by p, q, r

Q3

1. $d = e^{-1} \text{ mod } (p-1)(q-1)(r-1)$.
2. $x^{ed} - x \equiv 0 mod N$
3. $x(x^{ed-1} - 1)$ must be divisible by p, q, r
4. $x(x^{k(p-1)(q-1)(r-1)+1-1} - 1)$ must be divisible by p, q, r
5. $x(x^{k(p-1)(q-1)(r-1)} - 1)$ must be divisible by p, q, r

Q3

1. $d = e^{-1} \bmod (p-1)(q-1)(r-1)$.
2. $x^{ed} - x \equiv 0 mod N$
3. $x(x^{ed-1} - 1)$ must be divisible by p, q, r
4. $x(x^{k(p-1)(q-1)(r-1)+1-1} - 1)$ must be divisible by p, q, r
5. $x(x^{k(p-1)(q-1)(r-1)} - 1)$ must be divisible by p, q, r
6. Lets show that the above is divisible by p.

Q3

1. $d = e^{-1} \bmod (p-1)(q-1)(r-1)$.
2. $x^{ed} - x \equiv 0 mod N$
3. $x(x^{ed-1} - 1)$ must be divisible by p, q, r
4. $x(x^{k(p-1)(q-1)(r-1)+1-1} - 1)$ must be divisible by p, q, r
5. $x(x^{k(p-1)(q-1)(r-1)} - 1)$ must be divisible by p, q, r
6. Lets show that the above is divisible by p.
7. 2 cases. If x is divisible by p, we are done.

Q3

1. $d = e^{-1} \bmod (p-1)(q-1)(r-1)$.
2. $x^{ed} - x \equiv 0 mod N$
3. $x(x^{ed-1} - 1)$ must be divisible by p, q, r
4. $x(x^{k(p-1)(q-1)(r-1)+1-1} - 1)$ must be divisible by p, q, r
5. $x(x^{k(p-1)(q-1)(r-1)} - 1)$ must be divisible by p, q, r
6. Lets show that the above is divisible by p.
7. 2 cases. If x is divisible by p, we are done.
8. If not, then use Fermat's Little Theorum on the right term with mod p

Q3

1. $d = e^{-1} \bmod (p-1)(q-1)(r-1)$.
2. $x^{ed} - x \equiv 0 mod N$
3. $x(x^{ed-1} - 1)$ must be divisible by p, q, r
4. $x(x^{k(p-1)(q-1)(r-1)+1-1} - 1)$ must be divisible by p, q, r
5. $x(x^{k(p-1)(q-1)(r-1)} - 1)$ must be divisible by p, q, r
6. Lets show that the above is divisible by p.
7. 2 cases. If x is divisible by p, we are done.
8. If not, then use Fermat's Little Theorum on the right term with mod p
9. $(x^{p-1})^{k(q-1)(r-1)} - 1 \equiv 1 - 1 \equiv 0 \bmod p$.

Q3

1. $d = e^{-1} \bmod (p-1)(q-1)(r-1)$.
2. $x^{ed} - x \equiv 0 mod N$
3. $x(x^{ed-1} - 1)$ must be divisible by p, q, r
4. $x(x^{k(p-1)(q-1)(r-1)+1-1} - 1)$ must be divisible by p, q, r
5. $x(x^{k(p-1)(q-1)(r-1)} - 1)$ must be divisible by p, q, r
6. Lets show that the above is divisible by p.
7. 2 cases. If x is divisible by p, we are done.
8. If not, then use Fermat's Little Theorum on the right term with mod p
9. $(x^{p-1})^{k(q-1)(r-1)} - 1 \equiv 1 - 1 \equiv 0 \bmod p$.
10. Thus the eqn is divisible by p. Do the same for q, r.

Q3

1. $d = e^{-1} \bmod (p-1)(q-1)(r-1)$.
2. $x^{ed} - x \equiv 0 mod N$
3. $x(x^{ed-1} - 1)$ must be divisible by p, q, r
4. $x(x^{k(p-1)(q-1)(r-1)+1-1} - 1)$ must be divisible by p, q, r
5. $x(x^{k(p-1)(q-1)(r-1)} - 1)$ must be divisible by p, q, r
6. Lets show that the above is divisible by p.
7. 2 cases. If x is divisible by p, we are done.
8. If not, then use Fermat's Little Theorum on the right term with mod p
9. $(x^{p-1})^{k(q-1)(r-1)} - 1 \equiv 1 - 1 \equiv 0 \bmod p$.
10. Thus the eqn is divisible by p. Do the same for q, r.
11. Fact: $x^{\phi N} \equiv 1 \bmod N$