RSA Attacks

Michael Levin

Computer Science Department, Higher School of Economics

Outline

Simple Attacks

Small Difference

Insufficient Randomness

Hastad's Broadcast Attack

- Alice wants to secretly transmit “attack” or “don’t attack” to Bob

- Alice wants to secretly transmit “attack” or “don’t attack” to Bob
- Converts “attack” to message $m = 1$, “don’t attack” to message $m = 0$

- Alice wants to secretly transmit “attack” or “don’t attack” to Bob
- Converts “attack” to message $m = 1$, “don’t attack” to message $m = 0$
- Encrypts m with RSA to get ciphertext c

- Alice wants to secretly transmit “attack” or “don’t attack” to Bob
- Converts “attack” to message $m = 1$, “don’t attack” to message $m = 0$
- Encrypts m with RSA to get ciphertext c
- Unfortunately, it is easy to break the cipher: just encrypt both $m = 0$ and $m = 1$ with RSA and check which one results in c

- Alice wants to secretly transmit “attack” or “don’t attack” to Bob
- Converts “attack” to message $m = 1$, “don’t attack” to message $m = 0$
- Encrypts m with RSA to get ciphertext c
- Unfortunately, it is easy to break the cipher: just encrypt both $m = 0$ and $m = 1$ with RSA and check which one results in c
- Works with any small set of possible messages

Solution

- To solve this common problem, use randomness
- For example, use the first 128 bits for the message and append 128 more random bits before encryption
- Bob will be able to read the first 128 bits, and this simple attack won't work: more than 2^{128} possible messages

Small Prime

- Bob generates two random primes p and q

Small Prime

- Bob generates two random primes p and q
- What if one of them, p, is less than $1\ 000\ 000$?

Small Prime

- Bob generates two random primes p and q
- What if one of them, p, is less than $1\ 000\ 000$?
- Eve can try all primes up to $1\ 000\ 000$ as divisors of the public key n

Small Prime

- Bob generates two random primes p and q
- What if one of them, p, is less than $1\ 000\ 000$?
- Eve can try all primes up to $1\ 000\ 000$ as divisors of the public key n
- Factorize n and decrypt the cipher the same way Bob does

Small Prime

- Bob generates two random primes p and q
- What if one of them, p, is less than $1\ 000\ 000$?
- Eve can try all primes up to $1\ 000\ 000$ as divisors of the public key n
- Factorize n and decrypt the cipher the same way Bob does
- One typical solution is to generate random primes for the secret key uniformly among very large, 2048-bit numbers

Outline

Simple Attacks

Small Difference

Insufficient Randomness

Hastad's Broadcast Attack

Small Difference

- Bob generates p and q such that $p < q$ and the difference $r = q - p$ is small

Small Difference

- Bob generates p and q such that $p < q$ and the difference $r = q - p$ is small
- What can Eve do?

Small Difference

- $n = pq, p < q \Rightarrow p < \sqrt{n} < q$

Small Difference

- $n = pq, p < q \Rightarrow p < \sqrt{n} < q$
- $\sqrt{n} - p < q - p = r \Rightarrow \sqrt{n} - r < p < \sqrt{n}$

Small Difference

- $n = pq, p < q \Rightarrow p < \sqrt{n} < q$
- $\sqrt{n} - p < q - p = r \Rightarrow \sqrt{n} - r < p < \sqrt{n}$
- Try all integers between $\sqrt{n} - r$ and $\sqrt{n}$ as divisors of n

Small Difference

- $n = pq, p < q \Rightarrow p < \sqrt{n} < q$
- $\sqrt{n} - p < q - p = r \Rightarrow \sqrt{n} - r < p < \sqrt{n}$
- Try all integers between $\sqrt{n} - r$ and $\sqrt{n}$ as divisors of n
- Factorize n and decrypt the same way as Bob does

Even Faster

- p and q are both odd, so $\frac{p+q}{2}$ and $\frac{p-q}{2}$ are integers
- $n = pq = (\frac{p+q}{2} + \frac{p-q}{2})(\frac{p+q}{2} - \frac{p-q}{2}) = (\frac{p+q}{2})^2 - (\frac{p-q}{2})^2$
- So n is a difference of squares, and one of the squares is small, because $|p - q|$ is small
- We can try adding increasing squares of integers to n until we get an exact square of integer

Solution

- Generate p and q
- If $|p-q|$ is small, regenerate
- Repeat until $|p-q|$ is sufficiently large

Outline

Simple Attacks

Small Difference

Insufficient Randomness

Hastad's Broadcast Attack

- Attack by Heninger et al. and Lenstra et al.
- Use public keys from different devices!
- Experiment resulted in 0.4% factored HTTPS keys!

OpenSSL RSA key generation:

```
rng = RandomNumberGenerator()
rng.seed(seed)
p = rng.big_random_prime()
rng.add_randomness(bits)
q = rng.big_random_prime()
n = p * q
```

What if the `seed` is not random enough?

Example: keys are generated by the router immediately after startup, no incoming network packets to get randomness from yet.

OpenSSL RSA key generation:

```
rng = RandomNumberGenerator()
rng.seed(seed)
p = rng.big_random_prime()
rng.add_randomness(bits)
q = rng.big_random_prime()
n = p * q
```

What if the `seed` is not random enough?
Example: keys are generated by the router immediately after startup, no incoming network packets to get randomness from yet.

Sometimes the same p will be generated, with different q

Combine Public Keys

- If the public keys n_1 and n_2 are generated using the same p, but different q, then $\mathrm{GCD}(n_1, n_2) = p$, and we can factorize both n_1 and n_2.

Combine Public Keys

- If the public keys n_1 and n_2 are generated using the same p, but different q, then $\mathrm{GCD}(n_1, n_2) = p$, and we can factorize both n_1 and n_2.
- Take keys from many routers and try to combine all pairs

Combine Public Keys

- If the public keys n_1 and n_2 are generated using the same p, but different q, then $\mathrm{GCD}(n_1, n_2) = p$, and we can factorize both n_1 and n_2.
- Take keys from many routers and try to combine all pairs
- Make sure the random number generator is properly seeded

Combine Public Keys

- If the public keys n_1 and n_2 are generated using the same p, but different q, then $\mathrm{GCD}(n_1, n_2) = p$, and we can factorize both n_1 and n_2.
- Take keys from many routers and try to combine all pairs
- Make sure the random number generator is properly seeded
- Some computer programs ask the user to move mouse for some time to get randomness

Outline

Simple Attacks

Small Difference

Insufficient Randomness

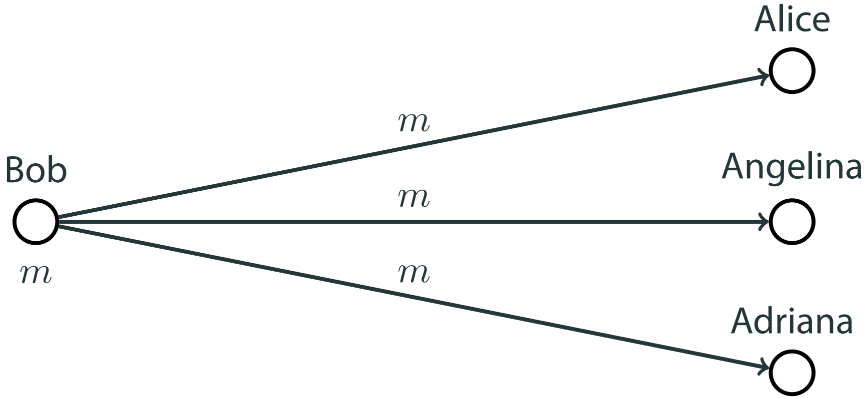
Hastad's Broadcast Attack

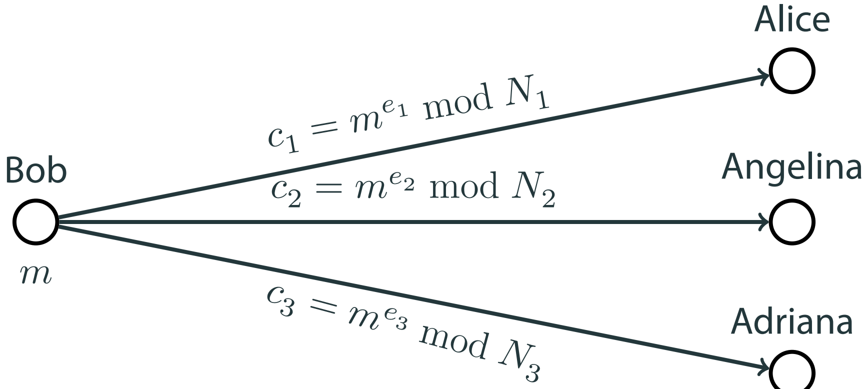
Hastad's Broadcast Attack

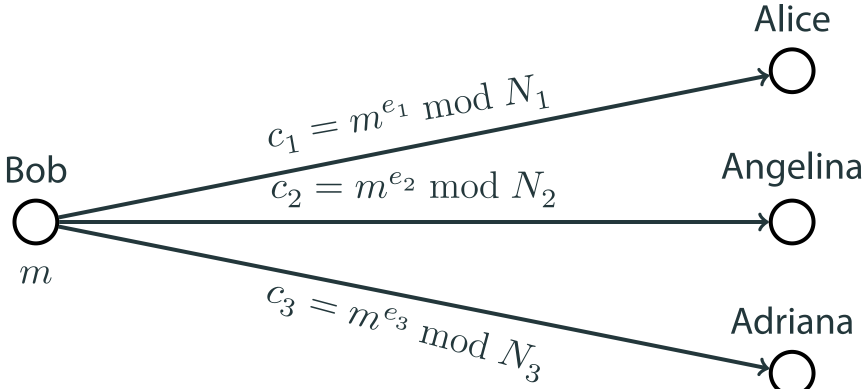
- Hastad came up with an attack in case Bob sends the same message m to several recipients using their public keys
- Uses the fact that the same message m is sent using different keys
- We will consider a very simplified case as an example


Alice
Bob
Angelina
Adriana

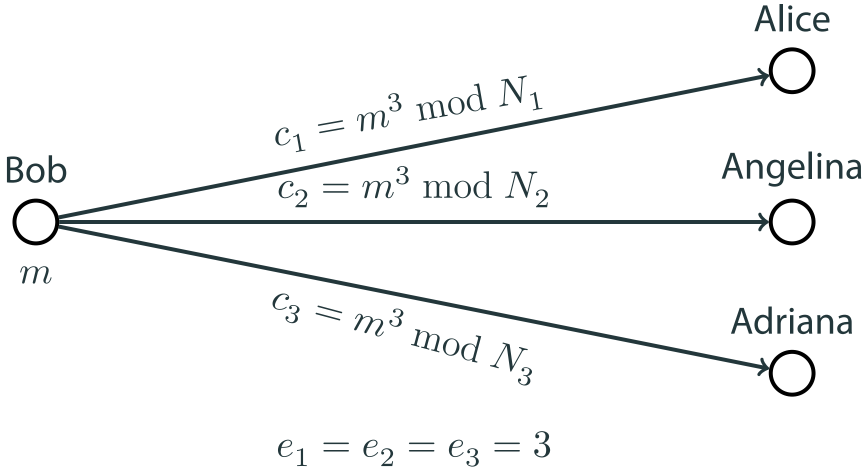

Alice
Bob
Angelina
m
Adriana


Alice
Angelina
Adriana
Bob
m
m
m
m


Alice
Angelina
Adriana
Bob
m
$c_1 = m^{e_1} \bmod N_1$
$c_2 = m^{e_2} \bmod N_2$
$c_3 = m^{e_3} \bmod N_3$


Alice
Angelina
Adriana
Bob
m
$c_1 = m^{e_1} \bmod N_1$
$c_2 = m^{e_2} \bmod N_2$
$c_3 = m^{e_3} \bmod N_3$

$$e_1 = e_2 = e_3 = 3$$


Alice
Angelina
Adriana
Bob
m
$c_1 = m^3 \bmod N_1$
$c_2 = m^3 \bmod N_2$
$c_3 = m^3 \bmod N_3$
$e_1 = e_2 = e_3 = 3$

$$c_1 \equiv m^3 \bmod N_1, c_2 \equiv m^3 \bmod N_2, c_3 \equiv m^3 \bmod N_3$$

$c_1 \equiv m^3 \bmod N_1, c_2 \equiv m^3 \bmod N_2, c_3 \equiv m^3 \bmod N_3$

$\mathrm{GCD}(N_i, N_j) = 1$, otherwise Eve can factor N_i and N_j to decrypt as in the previous attack.

$$c_1 \equiv m^3 \bmod N_1, c_2 \equiv m^3 \bmod N_2, c_3 \equiv m^3 \bmod N_3$$

$\mathrm{GCD}(N_i, N_j) = 1$, otherwise Eve can factor N_i and N_j to decrypt as in the previous attack.

Use Chinese Remainder Theorem to construct c such that $0 \leq c < N_1 N_2 N_3$ and

$$c \equiv c_1 \bmod N_1, c \equiv c_2 \bmod N_2, c \equiv c_3 \bmod N_3$$

$$c_1 \equiv m^3 \bmod N_1, c_2 \equiv m^3 \bmod N_2, c_3 \equiv m^3 \bmod N_3$$

$\mathrm{GCD}(N_i, N_j) = 1$, otherwise Eve can factor N_i and N_j to decrypt as in the previous attack.

Use Chinese Remainder Theorem to construct c such that $0 \leq c < N_1 N_2 N_3$ and

$$c \equiv c_1 \bmod N_1, c \equiv c_2 \bmod N_2, c \equiv c_3 \bmod N_3$$

Again by Chinese Remainder Theorem,

$$c \equiv m^3 \bmod N_1 N_2 N_3$$

$$c \equiv m^3 \bmod N_1 N_2 N_3$$

$$0 \leq c, m^3 < N_1 N_2 N_3$$

$$c \equiv m^3 \bmod N_1 N_2 N_3$$

$$0 \leq c, m^3 < N_1 N_2 N_3$$

So $c = m^3$

$$c \equiv m^3 \bmod N_1 N_2 N_3$$

$$0 \leq c, m^3 < N_1 N_2 N_3$$

So $c = m^3$

Eve can decode m as $m = \sqrt[3]{c}$

- Broadcasting the same fixed message is a problem
- Hastad's original attack works even with bigger and different e_i
- Solution — add random padding to m before encryption
- Then it is impossible to compute m using all c_i, because each c_i includes some randomness apart from m

More Attacks

- Time to compute $c^d \bmod n$ can expose d — if one can send ciphertexts to the server which decrypts them and sends some response
- Error return code in case of incorrect ciphertext can expose the message in the same case
- Power consumption while computing $c^d \bmod n$ can expose d — if one tries to decrypt an encrypted hard drive on a stolen computer, or withdraw cash from a stolen card using an ATM

Conclusion

- RSA is a powerful method which is used everywhere
- Hard to implement correctly, although the algorithm itself is relatively simple
- Attacks from unexpected angles
- Deeper dive in dedicated cryptography courses
- Have fun with the problems: let’s break some ciphers!