CSE 539: Applied Cryptography
Week 7: RSA

Ni Trieu (ASU)

Reading: https://joyofcryptography.com/pdf/chap13.pdf
https://en.wikipedia.org/wiki/RSA_(cryptosystem)

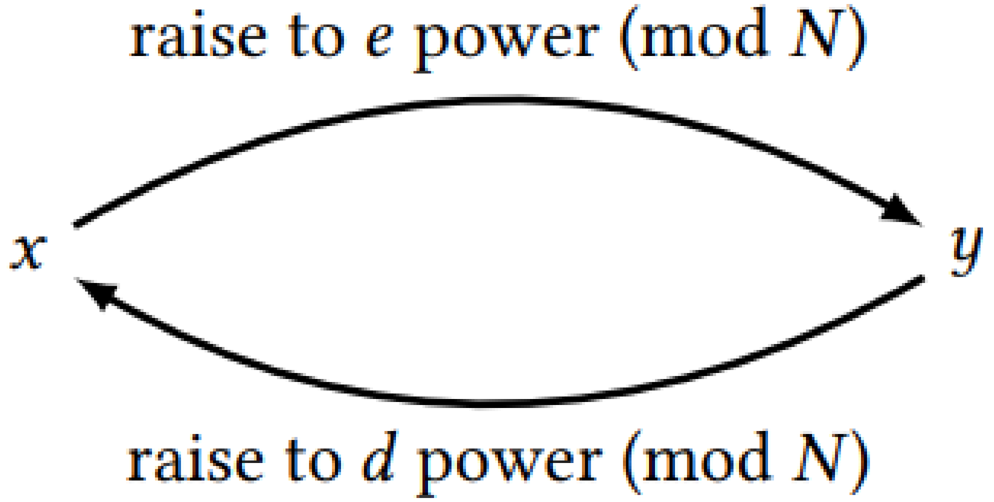
Recap: How RSA works

The RSA function is defined as follows:

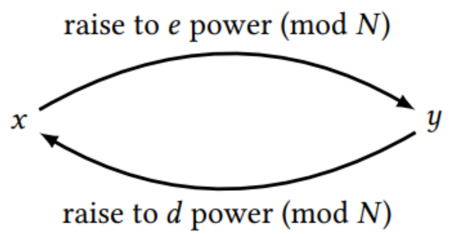
- Let p and q be distinct primes (later we will say more about how they are chosen), and let $N = pq$. N is called the **RSA modulus**.
- Let e and d be integers such that $ed \equiv_{\phi(N)} 1$. That is, e and d are multiplicative inverses mod $\phi(N)$ – not mod N!
- The RSA function is: $x \mapsto x^e \% N$, where $x \in \mathbb{Z}_N$.
- The inverse RSA function is: $y \mapsto y^d \% N$, where $x \in \mathbb{Z}_N$.

RSA Security

- Given only the public information (N, e), it should be hard to compute the RSA inverse (y -> y^d mod N) on randomly chosen values.
  - In other words, the only person who is able to compute the RSA inverse function is the person who generated the RSA parameters



RSA Security



- Currently the best known attacks against RSA (i.e., ways to compute the inverse RSA function given only the public information) involve factoring the modulus

=> understand the SOTA for factoring large numbers

- “Trial division” method of factoring

- The fastest factoring algorithm today is called the Generalized Number Field Sieve (GNFS)
    - https://en.wikipedia.org/wiki/General_number_field_sieve
    - https://en.wikipedia.org/wiki/RSA_Factoring_Challenge

RSA Security

- Example, Sage can easily factor reasonably large numbers. Factoring the following 200-bit RSA modulus takes about ~10 seconds

```
sage: p = random_prime(2^100)
sage: q = random_prime(2^100)
sage: N = p*q
sage: factor(N)
206533721079613722225064934611 * 517582080563726621130111418123
```

- As of Febrary 2020, the largest RSA modulus that has been (publically) factored is a 829-bit modulus
  - https://en.wikipedia.org/wiki/RSA_numbers#RSA-250
- Current best practices suggest to use 2048- or 4096-bit RSA moduli, meaning that p and q are each 1024 or 2048 bits.

https://www.sagemath.org/

RSA number	Decimal digits	Binary digits	Cash prize offered	Factored on	Factored by
RSA100	100	330	US$1,000[8]	April 1, 1991[9]	Arjen K. Lenstra
RSA110	110	364	US$4,429[8]	April 14, 1992[9]	Arjen K. Lenstra and M.S. Manasse
RSA120	120	397	US$5,898[8]	July 9, 1993[10]	T. Denny *et al.*
RSA129 [a]	129	426	US$100	April 26, 1994[9]	Arjen K. Lenstra *et al.*
RSA130	130	430	US$14,527[8]	April 10, 1996	Arjen K. Lenstra *et al.*
RSA140	140	463	US$17,226	February 2, 1999	Herman te Riele *et al.*
RSA150	150	496		April 16, 2004	Kazumaro Aoki *et al.*
RSA155	155	512	US$9,383[8]	August 22, 1999	Herman te Riele *et al.*
RSA160	160	530		April 1, 2003	Jens Franke *et al.*, University of Bonn
RSA170 [b]	170	563		December 29, 2009	D. Bonenberger and M. Krone [c]
RSA576	174	576	US$10,000	December 3, 2003	Jens Franke *et al.*, University of Bonn
RSA180 [b]	180	596		May 8, 2010	S. A. Danilov and I. A. Popovyan, Moscow State University[11]
RSA190 [b]	190	629		November 8, 2010	A. Timofeev and I. A. Popovyan
RSA640	193	640	US$20,000	November 2, 2005	Jens Franke *et al.*, University of Bonn
RSA200 [b] ?	200	663		May 9, 2005	Jens Franke *et al.*, University of Bonn
RSA210 [b]	210	696		September 26, 2013[12]	Ryan Propper
RSA704 [b]	212	704	US$30,000	July 2, 2012	Shi Bai, Emmanuel Thomé and Paul Zimmermann
RSA220 [b]	220	729		May 13, 2016	S. Bai, P. Gaudry, A. Kruppa, E. Thomé and P. Zimmermann
RSA230 [b]	230	762		August 15, 2018	Samuel S. Gross, Noblis, Inc.
RSA232 [b]	232	768		February 17, 2020[13]	N. L. Zamarashkin, D. A. Zheltkov and S. A. Matveev.
RSA768 [b]	232	768	US$50,000	December 12, 2009	Thorsten Kleinjung *et al.*[14]
RSA240 [b]	240	795		Dec 2, 2019[15]	F. Boudot, P. Gaudry, A. Guillevic, N. Heninger, E. Thomé and P. Zimmermann
RSA250 [b]	250	829		Feb 28, 2020[16]	F. Boudot, P. Gaudry, A. Guillevic, N. Heninger, E. Thomé and P. Zimmermann

RSA Security

- But, if we know extra information about p and q, we can break RSA security
- For example, given $\delta = |p - q|$

RSA Security

- But, if we know extra information about p and q, we can break RSA security
- For example, given that p and q are close (e.g. $|p - q| < 1000$)