



Systems and Internet Infrastructure Security

Network and Security Research Center
Department of Computer Science and Engineering
Pennsylvania State University, University Park PA

Public-Key Cryptography and Attacks on RSA

Readings: Twenty Years of Attacks on the RSA Cryptosystem
by Dan Boneh
Presentation by Dave King

Public-Key Cryptography

- Symmetric cryptography: same key is used for encryption and decryption.
- Asymmetric cryptography: different keys used for encryption and decryption.
- Public-Key cryptography: an asymmetric cryptography scheme where the key used for encryption is made public.

History of Cryptography

- 1969, GCHQ: James Ellis proves the possibility of private-key generation over an public channel, but cannot find a way to implement it.
- 1972, Berkeley. Karp develops the theory of NP-Completeness.
- 1973, QCHQ: Clifford Cocks develops public-key cryptography after thinking about it overnight. His discovery is classified and goes unused.
- 1974, QCHQ: Malcolm J Williamson invents what is later becomes the Diffie-Hellman key exchange algorithm.

- 1976, Stanford: Diffie, Hellman, and Merkle independently invent (and make public) a secret-key generation algorithm.
- 1977, MIT: Rivest, Shamir, and Adleman, based on difficulty of factoring large primes.
- 1978, Stanford: Merkle-Hellman public-key cryptosystem, based on knapsack (broken by Adi Shamir in 1982).
- etc

- Many of the mathematical mechanisms for public-key cryptography were developed prior to their publication.
- However, their utility was unclear to those that developed them.
- “One-way functions” and complexity were not mathematically well-understood early on.

- The totient function, $\varphi(n)$, is the count of numbers k less than n such that k and n are relatively prime.
 - ▶ $\varphi(p) = p - 1$ for p prime
 - ▶ $\varphi(pq) = (p - 1)(q - 1)$ for p, q prime
- Euler's Theorem: $a^{\varphi(n)} = 1 \pmod n$

Encryption and Decryption

- Let p, q , be primes.
- Let $N = pq$ (typically 1024 bits)
- Let e, d such that $ed = 1 \pmod{(p-1)(q-1)}$.
- Let $0 \leq M < N$ be the message.
- Let $C = M^e \pmod N$ be the ciphertext.
- Then $C^d = M^{ed} = M \pmod N$
 - ▶ Result follows by Euler's Theorem, since $ed = 1 \pmod{\varphi(N)}$.
- **AND** make e public, so anyone can see it!

Why RSA works

- “Easy” to compute $M^e \bmod N$ and $C^d \bmod N$
- “Hard” to determine d , even given e and N !
- We think that $f(x) = x^e \bmod N$ is a one-way function. (We do not know this.)
- Factoring N is the same thing as revealing d : Let $\langle N, e \rangle$ be an RSA public key. Given the private key d , one efficiently factor of N , and given the factorization of N , one can efficiently determine d .

- Elementary Attacks
 - ▶ common modulus
 - ▶ blinding
- Attacks on the actual RSA cryptosystem
 - ▶ low private exponent
 - ▶ low public exponent
- Attacks of implementations of the RSA cryptosystem
 - ▶ timing attacks

- Poor configuration: A different N must be used for all users in a system, since if you know your own public, private key pair you can factor N .
- Blinding: can fool principals into providing signatures on any message M by multiplying by r^e , where r is random.

Low Private Exponent

- Wiener developed an attack that works effectively when d is sufficiently small.
- Let $N = pq$ with $q < p < 2q$.
- Let $d < (1/3)N^{1/4}$.
- Given $\langle N, e \rangle$ where $ed = 1 \pmod{\varphi(N)}$, then Marvin can efficiently recover d .
- The details of this are involved. But there are other ways of providing faster decryption, in particular the CRT method. (currently used by OpenSSL for modular exponentiation)

- To reduce encryption time, we might want to use a small e . This is not known to lead to any total breaks of RSA. Most attacks use Coppersmith's Theorem.
- Let N be an integer and f be a monic polynomial of degree d . Let $X = N^{(1/d) - \epsilon}$. Then there is an efficient way to determine all $|x_0| < X$ such that $f(x_0) = 0 \pmod N$.

- **Broadcast Attack:** Suppose Bob is sending a message M to k parties, each using the public exponent e . If $k \geq e$, then an attacker can determine M .
- Suppose before encrypting M and sending that to party P_i , Bob applies the polynomial f_i to M and encrypts $f_i(M)$.
- Hastad's attack: this is still vulnerable.
- To defend against this attack, use randomized padding!

- Franklin-Reiter: if e is low and Marvin has ciphertexts for M and $f(M)$ for some publicly-known polynomial f , then there is an attack to recover M and $f(M)$.
- Coppersmith's Short Pad Attack: Alice sends a message to Bob that has been randomly padded on the end to it. Marvin intercepts it and prevents it from reaching its destination. Alice sends the same message to Bob again with a different random padding on the end. Marvin can determine M given these two ciphertexts.

- Partial Key Exposure: if $e < \sqrt{N}$, then Marvin can determine all of d from knowing a percentage of the bits of d .
- Theorem: If N is n bits, then you only need $n/4$ bits of d to reveal all of d . (holds for all e)
- When $e = 3$, the cryptosystem leaks half of the bits of d !

Implementation Attacks

- Common method of modular exponentiation: repeated squaring.
- $z = M$
 $C = 1$
for $i = 0$ to n
 if $d_i = 1$ then $C = C * z \bmod N$
 $z = z^2 \bmod N$
end
- At the end, $C = M^d \bmod N$

- To determine d , Marvin generates a large number of random messages and observes how long it takes to compute their ciphertexts.
- Loosely, by seeing how much time it takes to compute everything combined with the physical specifications of the computational device it is possible to determine the bits of d .
- Once a quarter of the bits of d have been discovered, Marvin can factor N .
- Similar attacks exist for other implementations of modular exponentiation. Implementing blinding prevents this attack since Marvin no longer knows what message is being decrypted.

- Random Faults: invalid signatures (perhaps generated by hardware error or radiation) combined with the CRT method of doing modular multiplication make it easy for Marvin to factor N .
- Bleichenbacher's Attack: suppose a server expects the first 16 bits of an encrypted message to be some fixed value. Then a server rejecting badly-formatted messages lets Marvin factor N .
 - ▶ “rejecting” here just means Marvin can determine the difference in the server's behavior.

- RSA seems to be a secure public-key cryptosystem, despite our best efforts.
- Still no proof that it is theoretically safe.
- Most attacks on RSA come from poor configuration or bad implementations.

- The Prehistory of Public Key Cryptography: <http://www.cs.columbia.edu/~smb/nsam-160/>
- British Document Outlines Early Encryption Discovery. New York Times, December 24, 1997. <http://www.nytimes.com/library/cyber/week/122497encrypt.html>