6 RSA and public key cryptography

- So far, our symmetric ciphers required a single **private key** k, a secret shared between the communicating parties.

  That leaves the difficult task of how to establish such private keys over a medium like the internet.

- In **public key cryptosystems**, there are two keys k_e, k_d, one for encryption and one for decryption. Bob keeps k_d secret (from anyone else!) and shares k_e with the world. Alice (or anyone else) can then send an encrypted message to Bob using k_e. However, Bob is the only who can decrypt it using k_d.

  It is crucial that the key k_d cannot be (easily) constructed from k_e.

RSA is one the first public key cryptosystems.

- It was described by Ron Rivest, Adi Shamir, and Leonard Adleman in 1977. (Note the initials!)
- However, a similar system had already been developed in 1973 by Clifford Cocks for the UK intelligence agency GCHQ (classified until 1997). Even earlier, in 1970, his colleague James Ellis was likely the first to discover public key cryptography.

Example 145. Let us emphasize that it should be surprising that something like public key cryptography is even possible.

Imagine Alice, Bob and Eve setting at a table. Everything that is being said is heard by all three of them. The three have never met before and share no secrets. Should it be possible in these circumstances that Alice and Bob can share information without Eve also learning about it?

Public key cryptography makes exactly that possible!

(RSA encryption)

- Bob chooses secret primes p, q.
- Bob chooses e (and then computes d) such that $de \equiv 1 \pmod{(p-1)(q-1)}$.
- Bob makes $N = pq$ and e public. His (secret) private key is d.
- Alice encrypts $c = m^e \pmod N$.
- Bob decrypts $m = c^d \pmod N$.

Does decryption always work? What Bob computes is $c^d \equiv (m^e)^d = m^{de} \pmod N$. It follows from Euler's theorem and $de \equiv 1 \pmod{\phi(N)}$ that $m^{de} \equiv m \pmod{\phi(N)}$ for all invertible residues m. It is not quite so obvious that this actually works for all residues. We will prove this next time.

Is that really secure? Well, if implemented correctly (we will discuss potential issues) RSA has a good track record of being secure. Next class, we will actually prove that finding the secret key d is as difficult as factoring N (which is believed, but has not been proven, to be hard). On the other hand, it remains an important open problem whether knowing d is actually necessary to decrypt a given message.

Example 146. (homework) If $N = 77$, what is the smallest (positive) choice for e?

Solution. (final answers only) Technically, $e = 1$ works but then we wouldn't be encrypting at all.

Note that e must be invertible modulo $\phi(N) = 6 \cdot 10 = 60$. Hence, $e = 2, 3, 4, 5, 6$ are not allowed.

The smallest possible choice for e therefore is $e = 7$.

Armin Straub
straub@southalabama.edu

Example 147. Bob's public RSA key is $N=55$, $e=7$.

(a) Encrypt the message $m=8$ and send it to Bob.

(b) Determine Bob's secret private key d.

(c) You intercept the message $c=2$ from Alice to Bob. Decrypt it using the secret key.

Solution.

(a) The ciphertext is $c=m^e \pmod N$. Here, $c\equiv 8^7 \pmod{55}$
$8^2\equiv 9$, $8^4\equiv 9^2\equiv 26$. Hence, $8^7=8^4\cdot 8^2\cdot 8\equiv 26\cdot 9\cdot 8\equiv 2 \pmod{55}$. Hence, $c=2$.

(b) $N=5\cdot 11$, so that $\phi(N)=4\cdot 10=40$.
To find d, we compute $e^{-1} \pmod{40}$ using the extended Euclidean algorithm:

$$\begin{array}{rl} & \gcd(7,40) \\ = & \gcd(2,7) \\ = & 1 \end{array} \qquad \begin{array}{rcl} \boxed{40} & = & 6\cdot\boxed{7}-2 \\ \boxed{7} & = & 3\cdot\boxed{2}+1 \end{array}$$

Backtracking through this, we find that Bézout's identity takes the form

$$1=\boxed{7}-3\cdot\boxed{2}=\boxed{7}-3\cdot\left(6\cdot\boxed{7}-\boxed{40}\right)=-17\cdot\boxed{7}+3\cdot\boxed{40}.$$

Hence, $7^{-1}\equiv -17\equiv 23 \pmod{40}$ and, so, $d=23$.

(c) We need to compute $m=c^d \pmod N$, that is, $m=2^{23} \pmod{55}$.
$2^2=4$, $2^4=16$, $2^8\equiv 36\equiv -19$, $2^{16}\equiv 19^2\equiv 31 \pmod{55}$. Hence, $2^{23}=2^{16}\cdot 2^4\cdot 2^2\cdot 2\equiv 31\cdot 16\cdot 4\cdot 2\equiv 8 \pmod{55}$.
That is, $m=8$ (as we already knew from the first part).

Example 148. (homework) Bob's public RSA key is $N=77$, $e=13$.

(a) Encrypt the message $m=2$ and send it to Bob.

(b) Determine Bob's secret private key d.

(c) You intercept the message $c=2$ from Alice to Bob. Decrypt it using the secret key.

Solution. (final answers only) (a) $c=30$. (b) $d=37$. (c) $m=51$.

Example 149. Is it a problem that $m=1$ is always encrypted to $c=1$? (Likewise for $m=0$.)

Solution. Well, it would be a problem if we reply to questions using YES (say, 1) and NO (say, 0) and encrypt our reply. However, this would always be a terrible idea in any deterministic public key cryptosystem (that is, a system, in which a message gets encrypted in a single way)!

Why? That's because Eve can just encrypt both YES and NO (or any collection of expected messages) and see which matches the ciphertext she intercepted.

Important conclusion. We must not send messages taken from a small predictable set and encrypt them using a deterministic public key cryptosystem like RSA.

Once realized, this is easy to fix: for instance, Alice can just augment the plaintext with some random garbage in such a way that Bob can discard that garbage after decryption. This is done when RSA is used in practice.

Example 150. RSA is so cool! Why do we even care about, say, AES anymore?

Solution. RSA is certainly cool, but it is very slow (comparatively). As such, RSA is not practical for encrypting larger amounts of data. RSA is, however, perfect for sharing secret keys, which can then be used for encrypting data using, say, AES.