and $\gcd(m_i, m_j) = 1$ *for* $i \neq j$*, then there exists an* r *such that* $r = r_i \pmod{m_i}$ *for* $0 \leq i \leq n$*.*

PROOF: The proof is by counting. Distinct values of r, $0 \leq r < \Pi m_i$, represent distinct sequences. To see that, note that if $r = r' \pmod{m_i}$ for all i, then $m_i | (r - r')$ for all i, and so $(\Pi m_i)|(r - r')$ (since the m_i's are pairwise co-prime). So $r = r' \pmod{(\Pi m_i)}$, and so $r = r'$ if both $r, r' \in \{0, 1, \ldots, \Pi m_i\}$.

But the total number of sequences $r_0, \ldots, r_n$ such that (8.2) holds is Πm_i. Hence every such sequence must be a sequence of remainders of some r, $0 \leq r < \Pi m_i$. □

Note that the CRT can be stated in the language of group theory as follows:

$$\mathbb{Z}_{m_1 \cdot m_2 \cdot \ldots \cdot m_n} \cong \mathbb{Z}_{m_1} \times \mathbb{Z}_{m_2} \times \cdots \times \mathbb{Z}_{m_n}$$

where the m_i's are pairwise co-prime.

8.3 RSA

It is well known that Adam and Eve no longer trust each other[4]. Adam sets up a mechanism whereby he can receive and decode encoded messages from an arbitrary person—and no one else (Eve in particular) can read them. To this end, Adam advertises a function f, and *anyone* can compute $f(m)$ for *any* message m, but only Adam can *efficiently* compute m from $f(m)$ using the function g, where $g(f(m)) = m$.

Choose two odd primes p, q, and set $n = pq$. Choose $k \in \mathbb{Z}^*_{\phi(n)}$, $k > 1$. Advertise f, where $f(m) = m^k \pmod{n}$. Compute $l = k^{-1}$ (inverse of k in $\mathbb{Z}^*_{\phi(n)}$). Now $\langle n, k \rangle$ are public, and the key l is secret, and so is the function g, where $g(C) = C^l \pmod{n}$. (Note that $g(f(m)) = m^{kl} \pmod{n} = m$.)

Note that computing the inverse of k in $\mathbb{Z}^*_{\phi(n)}$, that is l, can be done in polytime using the extended Euclidean algorithm. Just observe that if $k \in \mathbb{Z}^*_{\phi(n)}$, then $\gcd(k, \phi(n)) = 1$, so $\exists s, t$ such that $sk + t\phi(n) = 1$, and further s, t can be chosen so that s is in $\mathbb{Z}^*_{\phi(n)}$ (first obtain any s, t from the extended Euclidean algorithm, and then just add to s the appropriate number of (positive or negative) multiples of $\phi(n)$ to place it in the set $\mathbb{Z}^*_{\phi(n)}$, and adjust t by the same number of multiples (of opposite sign)). Set $l := s$.

Obviously RSA relies on the hardness of factoring integers for its security; if we were able to factor n, we would obtain p, q, and hence $\phi(n) = \phi(pq) = (p - 1)(q - 1)$, and so we would be able to compute l.

[4]See Genesis 3:15.

The first question is: why $m^{kl} =_n m$? Observe that $kl = 1 + (-t)\phi(n)$, where $(-t) > 0$, and so $m^{kl} =_n m^{1+(-t)\phi(n)} =_n m \cdot (m^{\phi(n)})^{(-t)} =_n m$, because $m^{\phi(n)} =_n 1$. Note that this last statement does not follow directly from Euler's Theorem, because $m \in \mathbb{Z}_n$, and not necessarily in $\mathbb{Z}_n^*$; in fact m must be in $\mathbb{Z}_n - \{0, p, q, pq\}$, so we could insist that the messages m are small relative to n, so that $0 < m < \min\{p, q\}$—in fact, we break a large message into those small pieces. By Fermat's little theorem, we know that $m^{(p-1)} =_p 1$ and $m^{(q-1)} =_q 1$, so $m^{(p-1)(q-1)} =_p 1$ and $m^{(q-1)(p-1)} =_q 1$, thus $m^{\phi(n)} =_p 1$ and $m^{\phi(n)} =_q 1$. This means that $p|(m^{\phi(n)}-1)$ and $q|(m^{\phi(n)}-1)$, so, since p, q are distinct primes, it follows that $(pq)|(m^{\phi(n)} - 1)$, and so $m^{\phi(n)} =_n 1$.

The second questions is: how to select random primes? Two random primes are needed to find the public key $n = pq$ for the RSA[5] encryption scheme. It is a non-trivial problem, primarily because verifying the primality of a number is difficult. Here is how we go about it: we know by the prime number theorem that there are about $\pi(n) = n/\log n$ many primes $\leq n$. This means that there are $2^n/n$ primes among n-bit integers, roughly 1 in n, and these primes are fairly uniformly distributed. So we pick an integer at random, in a given range, and apply a primality testing algorithm to it, which in practice means the Rabin-Miller test[6]; see section 7.1.2, algorithm 7.1.2.

We now discuss very briefly two issues related to the security of RSA. The first one is that the primes p, q cannot be chosen "close" to each other. Note that

$$n = \left(\frac{p+q}{2}\right)^2 - \left(\frac{p-q}{2}\right)^2.$$

Since p, q are close, we know that $s := \frac{p-q}{2}$ is small, and $t := \frac{p+q}{2}$ is only slightly larger than $n^{\frac{1}{2}}$, and $t^2 - n = s^2$ is a perfect square. So we try the following candidate values for t:

$$\lceil n^{\frac{1}{2}} \rceil, \lceil n^{\frac{1}{2}} \rceil + 1, \lceil n^{\frac{1}{2}} \rceil + 2, \ldots$$

until $t^2 - n$ is a perfect square s^2. Clearly, if s is small, we will quickly find such a t, and then $p = t + s$ and $q = t - s$.

The second issue is the following: suppose that Eve can compute $\phi(n)$ from n. Then she can easily compute the primes p, q (of course, if she can

[5]RSA is named after the first letters of the last names of its inventors: Ron **R**ivest, Adi **S**hamir, and Leonard **A**dleman.

[6]The fact that this method of selecting primes works is attested by the fact that encryption packages such as GPG (`www.gnupg.org`) use it, and they work very well.

compute $\phi(n)$ she can directly compute l, and she does not need p, q). To see this note that $\phi(n) = \phi(pq) = (p-1)(q-1)$. Then,

$$\begin{aligned} p + q &= n - \phi(n) + 1 \\ pq &= n \end{aligned} \tag{8.3}$$

Note that

$$(x-p)(x-q) = x^2 - (p+q)x + pq = x^2 - (n - \phi(n) + 1)x + n,$$

so we can compute p, q by computing the roots of this last polynomial, and using the quadratic formula $x = (-b \pm \sqrt{b^2 - 4ac})/2a$, we obtain that p, q are

$$\frac{(n - \phi(n) + 1) \pm \sqrt{(n - \phi(n) + 1)^2 - 4n}}{2}.$$

Suppose that Eve is able to compute l from n and k. If Eve knows l, then she knows that whatever $\phi(n)$ is, it divides $kl - 1$, so she has equations (8.3) but with $\phi(n)$ in the first equation replaced by $(kl-1)/a$, for some a. There is a randomized polytime procedure to find the appropriate a, and obtain p, q, but we do not describe it here.

Thus, if Eve is able to factor then she can obviously break RSA; on the other hand, if Eve can break RSA (by computing l from n, k), then she would be able to factor in randomized polytime. Conceivably Eve could be able to break RSA *without* computing l, so this observation does not relate the security of RSA to factoring all that tightly.

8.4 The Isolation Lemma

A *weight function* over a finite set U is a mapping from U to the set of positive integers. We naturally extend any weight function over U to one on the power set $\mathcal{P}(U)$ as follows. For each $S \subseteq U$, the weight of S with respect to a weight function W, denoted $W(S)$, is $\Sigma_{x \in S} W(x)$. Let F be a nonempty family of nonempty subsets of U. Call a weight function W *good* for F if there is exactly one minimum-weight set in F with respect to W. Call W *bad* for F otherwise.

Lemma 8.4.1 (Isolation) *Let U be a finite set. Let $F_1, \ldots, F_m$ be families of nonempty subsets over U, and let $D = |U|$. Let $R > mD$, and let Z be the set of all weight functions whose weights are at most R. Let α, $0 < \alpha < 1$, be such that $\alpha > \frac{mD}{R}$. Then, more than $(1-\alpha)|Z|$ functions in Z are good for all $F_1, \ldots, F_m$.*