Proof that the RSA decryption algorithm works

From the key generation algorithm:

- p and q are prime numbers
- $n=p.q$
- $phi(n)$ is the totient of n, which can be computed as $n=(p-1).(q-1)$
  - this computation of $phi(n)$ comes from the factorization of n into a product of prime numbers
  - http://en.wikipedia.org/wiki/Euler%27s_totient_function
- $d.e = 1\ [phi(n)]$

Theorem: for all M in $[0;n[$, $(M^e)^d = M\ [n]$

Proof:

- Case 1: $gcd(M,n)=1$.
  - Euler's Theorem states that if a and n are coprime positive integers, then $a^{phi(n)} = 1\ [n]$
  - http://en.wikipedia.org/wiki/Euler%27s_theorem
  - since M and n are coprime positive integers, then $M^{phi(n)} = 1\ [n]$
  - $(M^e)^d = M^{ed} = M^{(1+k.phi(n))} = M.(M^{(phi(n))})^k = M.1 = M\ [n]$
- Case 2: $gcd(M,n)<>1$.
  - The Chinese Remainder Theorem states that if $gcd(p,q)=1$, then $x=y\ [p]$ and $x=y\ [q]$ implies that $x=y\ [p.q]$
  - Since $gcd(p,q)=1$, we will simply show that (1) $M^{ed}=M\ [p]$ and (2) $M^{ed}=M\ [q]$ (which will imply that $M^{ed} = M\ [p.q] = M[n]$)
    - Since $gcd(M,n)<>1$ and $n=p.q$, we have either $gcd(M,n)=p$ or $gcd(M,n)=q$. We assume without loss of generality that $gcd(M,n)=p$.
    - Since p is a divisor of M, $M = k.p = 0\ [p]$, and $(M^e)^d = M^{ed} = (k^{ed}).(p^{ed}) = 0\ [p]$ (the last equality comes from $p^{...} = 0\ [p]$)
    - Thus, $M^{ed} = M\ [p]$, and the first part is proven
    - $gcd(M,q)=1$, thus we can apply Euler's theorem to show that $M^{phi(q)}=1\ [q]$, with $phi(q)=q-1$ as q is prime
    - $(M^e)^d = M^{ed} = M^{(1+k.phi(n))} = M.M^{(k.(p-1).(q-1))} = M.(M^{(q-1)})^{(k.(p-1))} = M.1^{(k.(p-1))} = M\ [q]$, and the second part is proven (the second-to-last equality comes from $M^{(q-1)}=M^{phi(q)}=1\ [q]$ thanks to Euler's Theorem)
    - Both parts are proven, which completes the proof of the second case.