Offene Verschlüsselung – Ein Widerspruch?

Grundidee des RSA-Public-Key-Kryptosystem
(benannt nach Rivest-Shamir-Adleman 1977)

A) Installation des Systems:
Eine sog. Zertifizierungsstelle konstruiert zwei 'große' Primzahlen p, q (Stellenzahl etwa 100),
und bildet deren Produkt $N = p \cdot q$, eine 200-stellige Zahl.
Sie wählt nun einen Chiffrierschlüssel e (e=encrypting), dies ist eine beliebige 200-stellige Zahl teilerfremd zu $m = (p-1)(q-1)$.
Zu dieser Zahl berechnet die Zertifizierungsstelle den Dechiffrierschlüssel d (d=decrypting) mit der Eigenschaft $ed \equiv 1 \bmod m$ (siehe modulo-Arithmetik).
Dies sichert für alle 100-stelligen Zahlen a die Beziehung $a^{ed} \equiv a \bmod N$.
Nun werden N und e als öffentliche Schlüssel für einen bestimmten Empfänger bekanntgegeben.
d hält der Empfänger geheim.
p, q und m werden von der Zertifizierungsstelle vernichtet!

Alle im nachfolgend beschriebenen Chiffrierverfahren gemachten Rechnungen sind sog. Rechnungen modulo N (siehe Aushang links).

B) Chiffrierung:
Nachrichten werden in 100-stellige Zahlen a umgewandelt.
(Eine solche 100-stellige Dezimalzahl stellt ungefähr eine 300-Bit- bzw. knapp 40-Byte-Zahl dar, und umfasst damit knapp 40 Zeichen.)
Soll einem Empfänger E eine solche Nachricht a zugeschickt werden, so berechnet man mit dem *öffentlich bekannten* Schlüssel $e = e_E$ des Empfängers E als chiffrierte Nachricht die Potenz $a^e \bmod N$.
E dechiffriert diese mit dem nur ihm bekannten Schlüssel d: $(a^e)^d$ und erhält wegen $(a^e)^d = a^{ed} \equiv a \bmod N$ die Ausgangsnachricht zurück.
Auf diese Weise kann jeder an E eine verschlüsselte Nachricht senden, ohne je Kontakt mit ihm zu haben; lediglich der in einem 'Schlüsselbuch' (wie einem Telefonbuch) öffentlich gemachte Schlüssel e wird benötigt.

C) Digitale Signatur:
Man kann dasselbe System auch verwenden für eine elektronische Unterschrift:
Der Empfänger E übermittelt an den Absender A einen Kontrolltext a (oder es ist universell ein Kontrolltext vereinbart, etwa Name und Anschrift des Absenders).
Dieser wird von A mit seinem geheimen Schlüssel d 'dechiffriert' $a^d \bmod N$ und an E (zurück)gesandt.
E kann nun diese Chiffre dem öffentlich bekannten Schlüssel $e = e_A$ des vorgeblichen Senders A unterwerfen und erhält $(a^d)^e = a^{de} \equiv a \bmod N$, also den Quelltext a zurück und damit den Nachweis, dass A der Sender war.

D) Signierte Nachricht:
Man kann B) und C) kombinieren:
Sender A unterwirft eine Nachricht a zunächst seiner nur ihm bekannten 'Dechiffrierung': $a^{d_A} \bmod N$, und verschlüsselt dies dann mit dem öffentlichen Schlüssel e_E des Empfängers E. Die chiffrierte Nachricht ist also $a^{d_A e_E} \bmod N$.
Diese kann nun E dechiffrieren und erhält wieder $a^{d_A} \bmod N$ (ein noch immer unverständlicher Text).
Wendet er darauf den öffentlichen Schlüssel e_A des (vorgeblichen) Senders A an, so erhält er den Klartext $a^{d_A e_A} = a \bmod N$ zurück und weiß zugleich, dass die Nachricht von A stammen muss.

E) Sicherheit:
Ob dies ein sicheres Kryptosystem ist, hängt davon ob, wie schwierig es ist, aus dem Chiffrierschlüssel e den Dechiffrierschlüssel d zu bestimmen.
Die Zertifizierungsstelle konnte aus e den Dechiffrierschlüssel d berechnen – weil sie die beiden Primzahlen p, q und damit auch die Zahl $m = (p-1)(q-1)$ kennt.
Da $N = pq$ öffentlich bekannt ist, hängt alles davon ab, wie aufwendig es ist, aus N die beiden Primfaktoren p, q wieder zu gewinnen.
Das RSA-Verfahren ist also in demselben Maße als sicher anzusehen, wie es schwierig ist, die Zerlegung einer Zahl in zwei Primfaktoren in begrenzter Zeit durchzuführen.
Beim heutigen Kenntnisstand und mit der verfügbaren Hardware ist es zwar durchaus möglich, zwei 100-stellige Primzahlen p und q zu erzeugen, aber praktisch unmöglich, deren 200-stelliges Produkt N zu faktorisieren.
Die Multiplikation natürlicher Zahlen ist eine sog. *Einbahnstraßen-Funktion,* siehe linken Aushang.
Bei steigender Rechenkapazität kann man leicht von 200-stelligen zu 300- oder 400-stelligen Zahlen übergehen, ohne das Grundsystem ändern zu müssen.