RSA

Einleitung

Wurden Verschlüsselungstechniken lange Zeit hauptsächlich im militärischen Bereich genutzt, so hat sich die Nachfrage danach seit der Ausbreitung des Internets auch im zivilen Sektor beträchtlich erhöht: Besonders einleuchtend ist, dass bei der Abwicklung von Bankgeschäften über das Internet, sowohl die Abhörsicherheit einer Nachricht als auch eine eindeutige Identifizierung des Absenders (elektronische Unterschrift) gewährleistet werden müssen.

Der Verschlüsselungsalgorithmus RSA wurde von seinen Erfindern (Rivest, Shamir, Adleman) im Jahr 1977 entwickelt. Er gilt heute als das zuverlässigste Verfahren.
Text *(Als Beispiel sei der Buchstabe D gewählt.)* wird zuerst auf eine triviale Weise in Zahlen *(x_D)* umgewandelt, diese Zahlen werden nach einem raffinierten Verfahren (dem Algorithmus RSA) mit einer Funktion f abgebildet auf andere Zahlen *(im Beispiel $f(x_D)$)*. Nachdem diese Zahlen dann über einen öffentlichen Kanal an den Empfänger übermittelt worden sind, kann der Empfänger dann zur Entschlüsselung mittels der Umkehrfunktion f^{-1} wieder die Zahlenkombination des ursprünglichen Textes berechnen. *($x_D = f^{-1}(f(x_D))$)*. Im letzten (trivialen) Schritt werden diese Zahlen wieder in die entsprechenden Buchstaben zurückübersetzt *(Aus x_D wird wieder D.)*.

Die Leistung des Algorithmus besteht dabei darin, dass

1. es für eine unberechtigte Person praktisch unmöglich ist, aus der Zahl $f(x_D)$ die Zahl x_D zu berechnen (Dadurch bleibt die Nachricht vor Abhören geschützt.) und
2. es für den Empfänger auf eindeutige Weise möglich ist, den Absender der Nachricht zu identifizieren (Authentizität). Dadurch können Unbefugte eine Nachricht eines anderen Absenders nicht vortäuschen.

Der Algorithmus - ein Beispiel

Im Folgenden wird dieses Verfahren an einem einfachen Beispiel, dem Buchstaben D, demonstriert, indem die Funktionen f und f^{-1} definiert und ihre Eigenschaften untersucht werden.
Der Empfänger wählt zwei große Primzahlen p und q sowie eine zu (p-1)(q-1) teilerfremde Zahl e *(Um die Rechnungen möglichst einfach zu gestalten, wählen wir im Beispiel jedoch kleine Primzahlen. p=3, q=11, e=7),* bildet das Produkt n=pq *(n=33)* und veröffentlicht in einem allgemein zugänglichen Verzeichnis (Telefonbuch, Internet HomePage,…) seinen so gewählten Schlüssel: (n, e) *(33,7)*. Auf keinen Fall wird er die beiden Faktoren p und q von n bekanntgeben, denn die Schwierigkeit für Unbefugte, den Text zu entschlüsseln, beruht darauf, dass es bei **großen** Zahlen p und q praktisch unmöglich ist, aus dem Produkt n die Faktorisierung n=pq zu finden.

Der Absender kann für die erste Übersetzung seines Textes in Zahlen die ASCII-Tabelle mod 33 verwenden: *D -----> 68 mod 33 = 2 =: x_D*. Im zweiten Schritt, der eigentlichen Verschlüsselung, informiert sich der Absender über den Schlüssel seines Adressaten und benutzt dann die Funktion $f(x):= x^e \bmod n$. *Das ergibt in unserem Beispiel $f(68) = 2^7 \bmod 33 = 29$. Diese Zahl 29 kann nun über den öffentlichen Kanal an den Empfänger gesendet werden. Um aus 29 dann wieder die Zahl 68 berechnen zu können muss der Empfänger die Umkehrfunktion f^{-1} kennen. In unserem Beispiel lautet sie $f^{-1}(x) = x^3 \bmod 33$. Damit erhält der Empfänger: $f^{-1}(29) = 29^3 \bmod 33 = 2$. (*Ein effizientes Verfahren zur Berechnung der Funktionswerte wird später vorgestellt, wenn es bei großen Zahlen nötig ist). *Der Großbuchstabe der nach der ASCII-Tabelle mod 33 der Zahl 2 entspricht ist D, und damit ist die Entschlüsselung vollbracht.*

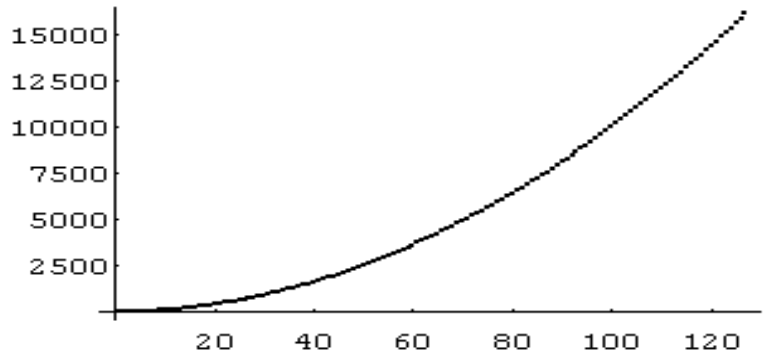
Aus diesem Beispiel ergeben sich eine Reihe von mathematischen Fragen, die auf den Kern des Verständnisses des RSA-Algorithmus abzielen:

1. Warum benutzt man für die Funktion f eine zahlentheoretische Funktion ?
2. Ist f immer umkehrbar?
3. Wie berechnet der Empfänger die Umkehrfunktion f^{-1} *(In unserem Beispiel musste der Empfänger den decodierenden Exponenten d= 3 finden.)* ?
4. Wie berechnet man effizient $f(x) = x^e \bmod n$ bei großen Zahlen x, e und n?
5. Warum ist es für einen Außenstehenden praktisch unmöglich, die Umkehrfunktion f^{-1} (bzw. die Zahl d) zu finden ?
6. Wie kann der Empfänger den Absender identifizieren?

Diese Fragen werden im Folgenden der Reihe nach beantwortet.

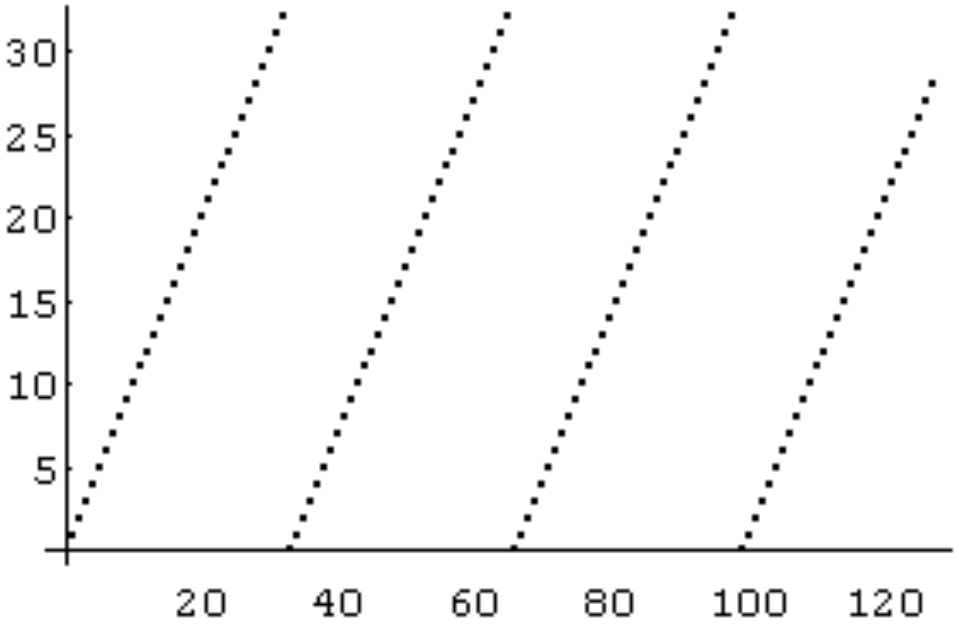
Warum benutzt man für die Funktion f eine zahlentheoretische Funktion ?

Nehmen wir an, der zu übermittelnde Text setzt sich aus den Zeichen der ASCII-Tabelle zusammen, dann liegt nach der entsprechenden Umwandlung jedes Zeichens in die der ASCII-Tabelle entsprechende Zahl die Zahlenmenge T vor, deren Elemente aus A:={0,1,2,..127} stammen. Die Verschlüsselungsfunktion f muss umkehrbar (Zwei verschiedene Elemente aus A müssen unter f verschiedene Bilder haben.) sein, damit eine spätere Entschlüsselung erfolgen kann. Funktionen mit dieser Eigenschaft gibt es viele:
$f(x) := x^2$ oder $f(x):= x^3$

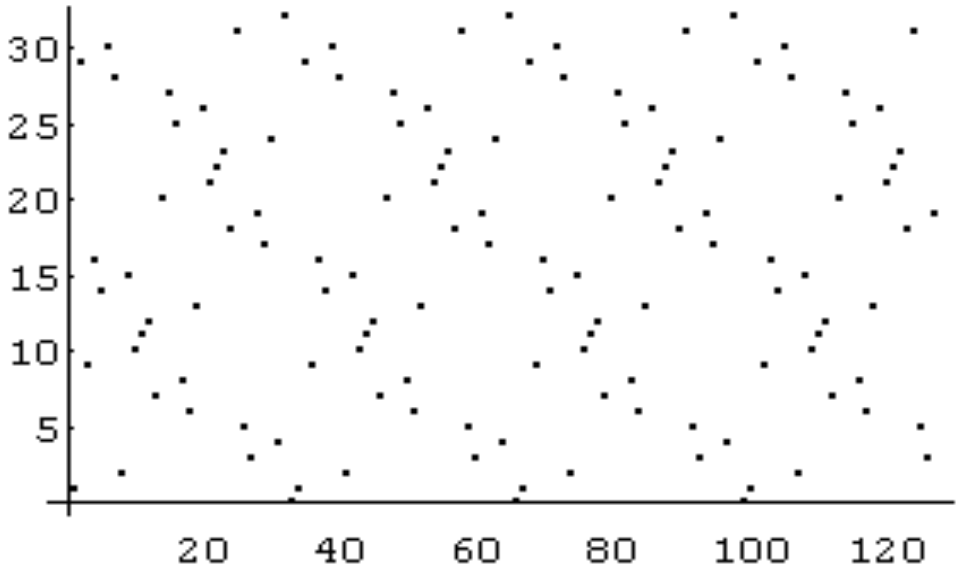


$$f(x) = x^2$$

Der Nachteil dieser Funktion ist offensichtlich ihre Monotonie, denn dadurch ist es für einen Unbefugten leicht möglich den, Code zu entschlüsseln: Wenn irgendwie bekannt ist, dass 4626 den Buchstaben D und 4900 den Buchstaben F (ASCII-Code 70) darstellt, dann ist klar, dass 4761 den Buchstaben darstellt, der dazwischen liegt. Das ist E. Also braucht man nicht monotone, umkehrbare Funktionen für die Verschlüsselung. Z.B. $f(x) = x \bmod 33$



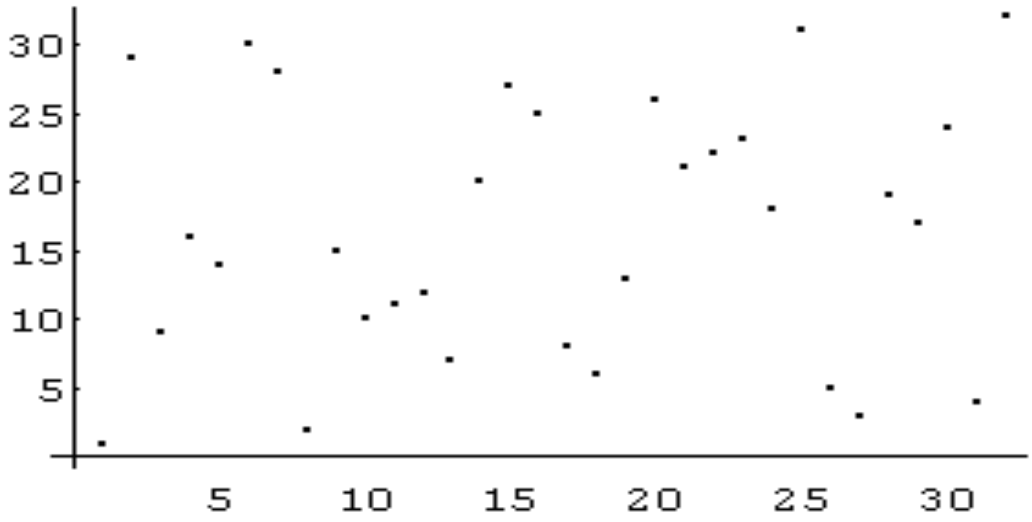
$f(x) = x \bmod 33$



Offensichtlich ist auch dieser Graph wegen seiner Regelmäßigkeit für eine Verschlüsselung unbrauchbar. Bessere Ergebnisse erzielt man, wenn man einen Exponenten einfügt: f(x) := x^7 mod 33

$$f(x) = x^7 \bmod 33$$

Selbstverständlich sind diese beiden Funktion nicht umkehrbar, weil der Modul 33 kleiner als 127 ist. Auf der Menge {0,1,2,…32} verhält sie sich besser:



Wenn also, wie es in der Praxis der Fall sein wird, große Primzahlen p und q gewählt werden, dann wird der Graph von f(x) = x^7 mod (pq) unregelmäßig, und anscheinend ist die Funktion auch umkehrbar.

Anmerkung: Der aufmerksame Leser wird vielleicht einwenden, dass, wie geschickt auch immer f gewählt sein mag, diese Funktion letztlich doch nur eine Zuordnung von der Menge A auf eine gleichmächtige andere Menge bewirkt. Durch eine Häufigkeitsanalyse in einem längeren, verschlüsselten Text könnte man dann schnell herausfinden, welche Zahlen den am häufigsten auftretenden Buchstaben e, a, d usw. eines deutschen Textes entsprechen. Damit wäre der Code geknackt. Diese Möglichkeit wird in der Realität ausgeschlossen, weil nicht wie im obigen Beispiel f(x_D) berechnet wird, sondern ganze Buchstabengruppen einer fest gewählten Länge werden nach der Umwandlung zu einer einzigen großen Zahl vor der Verschlüsselung zusammengefasst. Um die Beispielrechnungen nicht durch zu große Zahlen zu belasten, gehen wir auf diesen Aspekt nicht weiter ein.

Bevor wir zur weiteren Untersuchung der genannten Fragen kommen, soll ein einfacher Satz bewiesen werden, der das Rechnen mit Resten mod n erleichtert:
Satz: Für x,y, n ∈ N gilt: (x y) mod n = ((x mod n) (y mod n)) mod n.

Beweis: Es gibt positive Zahlen q_x und q_y mit den Eigenschaften x = q_x n + x mod n und y = q_y n + y mod n Dann gilt

$$
\begin{aligned}
(xy) \bmod n &= ((q_x n + x \bmod n)(q_y n + y \bmod n)) \bmod n \\
&= ((q_x q_y n + q_x (y \bmod n) + q_y (x \bmod n))n + (x \bmod n)(y \bmod n)) \bmod n \\
&= ((x \bmod n)(y \bmod n) \bmod n
\end{aligned}
$$

Ist f immer umkehrbar?

Für die folgenden Überlegungen benötigen wir einen Term, um die Anzahl der zu n teilerfremden Zahlen zu bestimmen.

Es sei $P_n := \{x \in \mathbb{N} | x < n$, und x und n sind teilerfremd$\}$ und $\varphi(n)$ bezeichne die Anzahl der Elemente in dieser Menge. Für unser Zahlenbeispiel bedeutet das: $P_{33} = \{1,2,4,5,7,8,10,13,14,16,17,19,20,23,25,26,28,29,31,32\}$, und $\varphi(33)=20$.

Satz: Für zwei verschiedene Primzahlen p und q gilt: $\varphi(pq) = (p-1)(q-1)$.

Beweis: Die Teiler von n sind die Vielfachen von p und von q, da dies die beiden einzigen Primfaktoren von n sind.

$$
\begin{aligned}
\varphi(n) &= \varphi(p\,q) \\
&= \text{alle zu pq teilerfremde Zahlen unterhalb von pq} \\
&= \text{alle Zahlen unterhalb von pq – Vielfache von q - Vielfache von p} \\
&= p\,q - 1 - (p-1)q - (q-1)p \\
&= pq - 1 - pq + q - pq + p \\
&= -1 + q - pq + p \\
&= -1 + q + p\,(1 - q) \\
&= (q-1)(p-1)
\end{aligned}
$$

Satz : Für jedes Element $x \in P_n$ und jede natürliche Zahl a gilt:

$$x^{\varphi(n)} \bmod n = 1 \text{ und } x^a \bmod n = x^{(a \bmod \varphi(n))} \bmod n .$$

Wir erläutern diesen Satz, seinen Beweis und seine Bedeutung an unserem Beispiel: $P_{33} = \{1,2,4,5,7,8,10,13,14,16,17,19,20,23,25,26,28,29,31,32\}$. Bezüglich der Multiplikation $\cdot \bmod 33$ bildet $\{P_{33}, \cdot\}$ eine abelsche Gruppe, weil das Produkt zweier zu 33 teilerfremden Zahlen natürlich wieder zu 33 teilerfremd ist[1]. Die Anzahl der Elemente in P_{33} ist gleich der Anzahl $\varphi(33)$ der zu 33 teilerfremden Zahlen: $\varphi(33)=20$. Wegen des Satzes von Lagrange[2] (Die kleinstmögliche positive Anzahl von Verknüpfungen, für die ein Element einer endlichen Gruppe mit sich selbst verknüpft das neutrale Element ergibt, ist ein Teiler der Anzahl der Gruppenelemente.) gilt für jedes $x \in P_{33}$: $x^{20} \bmod 33 = 1$.

---

[1] In der Zahlentheorie beweist man mit dem euklidischen Algorithmus, dass jedes Element aus P_n auch sein Inverses in P_n hat.

[2] Ein Beweis dieses elementaren Satzes der Gruppentheorie findet sich in jedem Lehrbuch zu diesem Thema.

Da die Zahl a darstellbar ist in der Form $a= q\,\varphi(n) + (a \bmod \varphi(n))$, für eine ganze Zahl q, folgt für jedes $x \in P_{33}$: $x^a \bmod 33 = x^{q\,\varphi(33) + (a \bmod \varphi(33))} \bmod 33$

$= (x^{q\,\varphi(33)} \bmod 33)\,(x^{a \bmod \varphi(33)} \bmod 33))$

$= (x^{\varphi(33)} \bmod 33)^q\,(x^{a \bmod \varphi(33)} \bmod 33))$

$= 1 \cdot x^{a \bmod \varphi(33)} \bmod 33).$

Der allgemeine Beweis des Satzes verläuft genauso.

Dieser Satz erlaubt es nun, die Gleichung $f(x_D)= x^e \bmod n$ (*) auf elegante Weise nach x_D aufzulösen: Wenn man eine Zahl $d \in P_n$ (d darf auch außerhalb von P_n liegen.)gefunden hat mit der Eigenschaft $(e\,d) \bmod \varphi(n) = 1$ (Wie man d berechnet wird anschließend erklärt.), dann potenziert man die Gleichung (*) mit d und erhält:

$(f(x_D))^d = (x_D^e \bmod n)^d$

$= (x_D^{e\,d}) \bmod n$

$= x_D^{ed} \bmod n$

$= x_D^{ed \bmod \varphi(n)} \bmod n$

$= x_D^1 \bmod n$

$= x_D$

*In unserem Beispiel benötigen wir eine Zahl d mit 7*d mod 20 = 1. Nach kurzer Suche findet man: 7*3 mod 20 = 1. Weil $f(x_D) = 29$, gilt also:$(29)^3 \bmod 33 = (-4)^3 \bmod 33 = 2$ und x_D ist erfolgreich berechnet.*

Nach einem bekannten Ergebnis der Zahlentheorie[3], gibt es immer dann zu einem Exponent e genau einen decodierenden Exponent d mit der Eigenschaft $(e\,d) \bmod \varphi(n) = 1$, wenn e und $\varphi(n)$ teilerfremd sind. *(Aus diesem Grund musste bei der Wahl von e=7 darauf geachtet werden, dass e teilerfremd zu 20 ist.)*

Nach dem obigen Satz ist die Umkehrbarkeit von f nur bewiesen für den Fall, dass Elemente aus P_n in die Funktion f eingesetzt werden. Wir beweisen nun noch, dass auch für die übrigen Argumente $x < n$ die Funktion f genauso umkehrbar ist wie auf der Menge P_n.

Ein Beispiel soll das erläutern: $6 \notin P_{33}$, weil 6 und 33 nicht teilerfremd sind. Dennoch gilt

$f(6) = 6^7 \bmod 33 = (6^2 \bmod 33)^3\ 6 \bmod 33 = 27 * 6 \bmod 33 = 30$; und

$f^{-1}(30) = 30^3 \bmod 33 = (-3)^3 \bmod 33 = -27 \bmod 33 = 6.$

[3] Vgl. Scholz, Schoeneberg: Einführung in die Zahlentheorie Sammlung Göschen 1973

Da ed mod φ(n) = 1, und φ(n) = (p-1)(q-1), folgt ed mod (p-1) = 1 und ed mod (q-1) = 1. Also gilt für eine natürliche Zahl x kleiner als n =pq : $x^{ed} = x \bmod p$ und $x^{ed} = x \bmod q$. Oder: (x^{ed}-x) ist duch p und durch q teilbar. Da p und q verschiedene Primzahlen sind, ist (x^{ed}-x) auch durch pq teilbar; das heißt: $x^{ed} \bmod n = x$

Zusammenfassend erhält man das Ergebnis: Seien n = pq, wobei p und q zwei verschiedene Primzahlen sind, dann ist $f(x):= x^e \bmod n$ umkehrbar auf der Menge {0,1,2,…n-1}, wenn e teilerfremd zu φ(n) ist. Die Gleichung der Umkehrfunktion f^{-1} lautet $f^{-1}(x):= x^d \bmod 33$, wobei d die Gleichung (e d) mod φ(n) = 1 erfüllen muss.

Wie berechnet der Empfänger die Umkehrfunktion f^{-1} ?

Nach dem obigen Ergebnis ist diese Frage nun gleichbedeutend mit der Frage nach der Berechnung von d, wenn p, q und e gegeben sind. Man beachte, dass hier die Kenntnis der Faktoren von n= pq benötigt wird.

Nur an dieser Stelle benötigt man die Zerlegung von n in seine Primfaktoren.

Für unser Beispiel p=3, q=11 und e=7 wird demonstriert, wie der invertierende Exponent d berechnet wird: Es gilt: φ(33) = (11-1)(3-1) = 20, und es ist die Gleichung e d mod φ(n) = 1 zu lösen. Für unser Beispiel: 7 d mod 20 = 1.

*20 = 1 * 20 + 0 * 7 I*
*7 = 0 * 20 + 1 * 7 II*

*I-2*II 6 = 1 * 20 +(-2) *7 III*

*II-III 1 = (-1) * 20 + 3 * 7*

*Ergebnis: 7*3 = 1 + 1* 20 oder 7*3 mod 20 = 1*

Bei größeren Zahlen p und q wird der Algorithmus deutlicher:
Seien p= 17, q= 23 und e=21 (e muss teilerfremd zu 16 und 22 sein.)

352 = 1 * 352 + 0 * 21 (I)
21 = 0 * 352 + 1 * 21 (II)

I – 16*II 16 = 1* 352 -16 * 21 (III)

II – 1*III 5 = -1 * 352 +17 * 21 (IV)

III –3*IV 1 = 4 * 352 - 67 * 21

Also: 21 * (-67) = 1 mod 352 oder 21 * 285 = 1 mod 352.

Dieses Verfahren läßt sich auch bei großen Zahlen leicht auf einem Computer programmieren.

Berechnung von $f(x) = x^e \bmod n$

Durch mehrfache Anwendung des eingangs bewiesenen Satzes zum Rechnen mit Resten, läßt sich $x^e \bmod n$ effizient berechnen, wenn man e in eine Summe von Zweierpotenzen zerlegt. Ein Beispiel soll dies illustrieren:

$$
\begin{aligned}
29^{23} \bmod 33 &= 29^{16+4+2+1} \bmod 33 \\
&= (29^{16}\, 29^{4}\, 29^{2}\, 29^{1}) \bmod 33 \\
&= ((29^{4})^{4} \bmod 33)\ ((29^{2})^{2} \bmod 33)(29 \bmod 33)^{2}\ (29 \bmod 33) \bmod 33 \\
&= ((29^{4})^{4} \bmod 33)\ ((29^{2})^{2} \bmod 33)\ 16\ (-4) \bmod 33 \\
&= ((29^{4})^{4} \bmod 33)\ (((29 \bmod 33)^{2})^{2} \bmod 33)\ 16\ (-4) \bmod 33 \\
&= ((29^{4})^{4} \bmod 33)\ ((16)^{2} \bmod 33)\ 16\ (-4) \bmod 33 \\
&= ((29^{4})^{4} \bmod 33)\ (256 \bmod 33)\ 16\ (-4) \bmod 33 \\
&= ((29^{4})^{4} \bmod 33)\ 25\ 16\ (-4) \bmod 33 \\
&= (25^{4} \bmod 33)\ (-8)\ 2 \bmod 33 \\
&= (-8)^{4} \bmod 33\ (-16) \bmod 33 \\
&= (64 \bmod 33)^{2}\ (17) \bmod 33 \\
&= (-2)^{2}\ (17) \bmod 33 \\
&= 68 \bmod 33 \\
&= 2
\end{aligned}
$$

Dieses Verfahren ist auch dann noch auf einem Rechner in kurzer Zeit durchführbar, wenn alle beteiligten Zahlen groß sind.

Warum ist es für einen Außenstehenden praktisch unmöglich, die Umkehrfunktion f^{-1} (bzw. die Zahl d) zu finden ?

Wenden wir uns nun der realistischen Situation zu, in der vom Empfänger der Nachricht große Primzahlen für p und q gewählt werden. Zunächst sei bemerkt, dass es in der Zahlentheorie leistungsfähige Algorithmen gibt, nach denen ein Computer in kurzer Zeit Primzahlen mit sehr großer Stellenzahl berechnen kann. Wenn p und q nun beide etwa 100 Stellen im Dezimalsystem haben, dann hat das Produkt n = pq etwa 200 Stellen. Ein Außenstehender, der allein aufgrund der Kenntnis des veröffentlichen Schlüssels (n, e) den decodierenden Exponenten d bestimmen will, muss die Gleichung $e*d \bmod \varphi(n) = 1$ lösen. Also muss er $\varphi(n)$ berechnen. Für den richtigen Empfänger der Nachricht ist das einfach:
$\varphi(n) = (p-1)(q-1)$, aber für einen Außenstehenden ist es ohne die Kenntnis der Primfaktorzerlegung von n fast nur durch Probieren (auf einem Computer) möglich, entweder die zu n teilerfremden Zahlen alle zu berechnen und ihre Anzahl zu bestimmen,

oder die Faktoren von n durch Probieren zu suchen. Ein effizienter Algorithmus zur Primfaktorzerlegung einer großen Zahl ist nicht bekannt.

Ein einfaches Rechenbeispiel demonstriert die Aussichtslosigkeit für einen Außenstehenden, durch bloßes Probieren die Primfaktoren von n zu finden:

Wir nehmen vereinfachend an, ein Computer könne in einer Sekunde für 10^8 aufeinanderfolgende Zahlen untersuchen, ob sie Faktoren von n sind. (Ein üblicher PC mit einer Taktfrequenz von 500 MHz schafft höchstens ein Zehntel von dieser Anzahl.) Um von n dann den ersten Primfaktor, der vermutlich 100 Stellen hat, zu finden benötigt der Rechner dann etwa $10^{100}/10^8 = 10^{92}$ Sekunden. Das Alter des Universums wird auf etwa $15*10^9$ Jahre geschätzt; das sind (nur!) $15*10^9*365*24*60*60$ Sekunden $\approx 4{,}71*10^{17}$ Sekunden.

Wie kann der Empfänger den Absender identifizieren?

Der Absender der Nachricht muß, um sich ausweisen zu können, über einen eigenen veröffentlichten Schlüssel (n_A,e_A) verfügen, der dem Empfänger bekannt ist. Dann kann nämlich der Absender seine Nachricht einmal wie oben schon beschrieben mit dem Schlüssel (n,e) des Empfängers verschlüsseln und zusätzlich verschlüsselt er seinen Originaltext mit seinem geheimen Schlüssel (n_A,d_A) und dann weiter mit dem Schlüssel (n, e). An den Empfänger verschickt er nun über den öffentlichen Kanal beide Pakete. Dieser entschlüsselt beide Pakete mit seinem geheimen Schlüssel (n,d). Das erste Paket enthält die Nachricht im Klartext, und das zweite ist noch mit dem Schlüssel (n_A,d_A) verschlüsselt. Nun entschlüsselt der Empfänger das zweite Paket mit dem öffentlichen Schlüssel (n_A,e_A) des Absenders (Dazu beachte man, dass die Rollen des codierenden Exponenten und des decodierenden Exponenten vertauscht werden können.). Wenn nun auch das zweite Paket im Klartext mit dem ersten übereinstimmt, dann stammt die Nachricht vom richtigen Absender. Für einen Außenstehenden ist es nicht möglich, eine Nachricht des Absenders vorzutäuschen, weil die Zahl d_A nur dem Absender bekannt ist. Nur wenn eine Nachricht mit (n_A,d_A) verschlüsselt wurde, ergibt sie bei der Entschlüsselung mit (n_A,e_A) wieder Klartext.