

1. Problemstellung

Die Sicherheit des RSA-Verfahrens [1,2] ist bestimmt durch den algorithmischen und damit rechentechnischen Schwierigkeitsgrad der Refaktorisierung des Moduls n , des Produktes zweier genügend großer nichtspezieller Primzahlen $prime(1)$ und $prime(2)$

$$n = prime(1) \cdot prime(2) . \tag{1}$$

Die kommerzielle Bedeutung dieses einfach formulierbaren Problems: **Faktorisierung einer Semiprimzahl Gl. (1)** [3] und der Stand seiner Lösung sind nicht zuletzt an der Ausschreibung eines hoch dotierten Faktorisierungspreises für spezielle „RSA challenge numbers“ durch die amerikanische Firma RSA Security ablesbar [4,5,6], die damit den RSA-Sicherheitsstandard ständig auf den Prüfstand aller denkbaren Codebreaker-Ideen stellt, wohl aber auch unausgesprochen die Suche nach neuen public key cryptography Verfahren stimulieren will.

RSA- d challenge number mit d Dezimalstellen	Für die Faktorisierung ausgesetzter Preis in T \$	Stand der Faktorisierung
100		April 1991 faktorisiert
110		April 1992 faktorisiert
120		Juni 1993 faktorisiert
129		April 1994 faktorisiert
130		April 1996 faktorisiert
140		Februar 1999 faktorisiert
150		April 2004 faktorisiert
155		Aug. 1999 faktorisiert
160		April 2003 faktorisiert
174	10	Dezember 2003 faktorisiert
193	20	
200		Mai 2005 faktorisiert
212	30	noch nicht faktorisiert
232	50	noch nicht faktorisiert
270	75	noch nicht faktorisiert
309	100	noch nicht faktorisiert
463	150	noch nicht faktorisiert
617	200	noch nicht faktorisiert

Die hier benutzten Algorithmen sind zahlentheoretisch basiert [1,2,3,4,5,6], die erforderlichen Rechnerleistungen hoch, die Rechenzeiten lang (RSA-200: Faktorisierungsstart Dezember 2003, Ende: Mai 2005), die klassischen Ansätze scheinen ausgeschöpft. Die Suche nach alternativen Faktorisierungsalgorithmen ist daher naheliegend. Im folgenden soll ein Ansatz für einen solchen engineeringbasierten Algorithmus zur Diskussion gestellt werden.

2. Ein schneller Algorithmus zur Zerlegung einer positiven reellen Zahl in zwei Faktoren

Die Outputsequenz $y(0), y(1), y(2), y(3), \dots$ der Rekursionsgleichung

$$y(t+1) = y(t-1) \frac{1 + 3 \left(\frac{a}{y(t-1) \cdot y(t)} \right)^2}{3 + \left(\frac{a}{y(t-1) \cdot y(t)} \right)^2} \quad (2)$$

zeigt für positive Startwerte

$$y(0), y(1) \in \mathbb{R}_+ \quad (3)$$

und konstanten Input

$$x(t) = a \in \mathbb{R}_+ \quad (4)$$

nach einem kurzen Einschwingvorgang (≈ 20 Rekursionsschritte) einen periodischen Verlauf mit der Periodendauer 2

$$\dots, f(1), f(2), f(1), f(2), \dots \quad (5)$$

Dabei ist

$$f(1) \cdot f(2) = a. \quad (6)$$

Mit anderen Worten: Die Rekursionsgleichung (2) realisiert im stationären Fall eine Zerlegung des konstanten Inputs a in zwei Faktoren $f(1)$ und $f(2)$. Die konkrete Zerlegung wird dabei durch die Startwerte $y(0)$ und $y(1)$ bestimmt und ist so steuerbar („Diode \rightarrow Transistor“).

Das bedeutet aber auch, dass jede von außen erfolgende „Injektion“ Δy in nur ein $y(t)$

$$y(inj, t) = y(t) + \Delta y \in \mathbb{R}_+ \quad (7)$$

von einer stationären Zerlegung von a über einen neu beginnenden Einschwingvorgang zu einer anderen stationären Zerlegung von a führt [7].

Das folgende Beispiel 1 demonstriert die Leistungsfähigkeit dieses Faktorisierungsalgorithmus (Gln. (2) – (6)).

Beispiel 1

$$a = 30; y(0) = 70; y(1) = 65$$

Abarbeitungsprotokoll:

t	$y(t)$	$dy(t) = \frac{y(t)-y(t+2)}{y(t+2)-y(t+4)}$
0	70.0	3.0135781451882043
1	65.0	3.1232271663789791
2	23.336038281456292547	4.1944765265562106
3	21.689264445289779694	22.7656830020804191
4	7.851468288455681506	6639.8964024243985559
5	7.821962053410289366	$1.4145524615155874 \cdot 10^{11}$
6	4.159810963646690135	$2.5127247785417172 \cdot 10^{33}$
7	7.212830212708991681	$1.5861625645225770 \cdot 10^{100}$
8	4.159254982483571224	
9	7.212830212704685500	
10	$4.159254982483571224 = f(1)$	
11	$7.212830212704685500 = f(2)$	

Aus Spalte 3 dieses Abarbeitungsprotokolls entnimmt man unmittelbar: Mit steigendem t gilt mit steigender Genauigkeit

$$dy(t+1) = (dy(t))^3, \quad (8)$$

d.h., die steuerbare Faktorisierung einer positiven reellen Zahl konvergiert kubisch! Ersetzt man in Gl. (2) das Produkt $y(t-1) \cdot y(t)$ durch die Summe $y(t-1) + y(t)$, so erfolgt in derselben Weise wie vorstehend beschrieben im stationären Fall eine steuerbare Zerlegung des konstanten Inputs a in zwei Summanden $g(1)$ und $g(2)$ mit $g(1) + g(2) = a$, diese Zerlegung konvergiert jedoch nur linear.

3. Eine Kombinationsidee: Lösung des Faktorisierungsproblems von Semiprimzahlen unter Benutzung von Problemlösungen aus der Synchrotrontechnologie [8,9]

Die in Abschnitt 2 beschriebenen experimentell gefundenen Fakten sind Grundlage für eine Kombinationsidee: Gl. (2) könnte als „Beschleunigungsstrecke“ und zugleich als „Fokussierungsmagnet“ in einem „semiprime factoring alternating gradient synchrotron“ Einsatz finden, das eine höhere Leistungsfähigkeit als klassische Faktorisierungsalgorithmen besitzt. Dies impliziert aber sofort die Notwendigkeit, über Elemente für einen zugehörigen „Defokussierungsmagneten“ nachzudenken.

Die einfachstdenkbare Konstruktionsidee hierfür ist eine Rekursionsgleichung mit gegenüber Gl. (2) abgewandelten Vorzeichen:

$$y(t+1) = y(t-1) \frac{-1 + 3 \left(\frac{a}{y(t-1) \cdot y(t)} \right)^2}{3 - \left(\frac{a}{y(t-1) \cdot y(t)} \right)^2}; \quad t = 1, 2, 3, \dots \quad (9)$$

Beispiel 2 zeigt ein typisches Verhaltensmuster der Outputsequenz von Gl. (9).

Beispiel 2

$$a = 30; \quad y(0) = 5; \quad y(1) = 7$$

Abarbeitungsprotokoll:

t	$y(t)$	t	$y(t)$
0	5.0	13	-15.2374828327
1	7.0	14	-4.8443495200
2	2.6576576576	15	2.7116016489
3	119.1573564584	16	32.0233615179
4	-0.8646216223	17	-0.6042529769
5	-30.4770655312	18	333.5208036712
6	-1.4656690361	19	0.1894269421
7	-4.2225017154	20	-38.8937295606
8	4.9690311468	21	-0.6798677435
9	-22.6809011887	22	-64.9766841335
10	-1.3358066689	23	-0.1027149858
11	-21.8035898462	24	225.1429060286
12	-1.5037902189		

Obwohl das Startprodukt $y(0) \cdot y(1) = 35$ hier (im Gegensatz zu Beispiel 1 mit $y(0) \cdot y(1) = 4550$) nahe bei $a = 30$ liegt, divergiert die Outputsequenz von Gl. (9) in Form von deterministischem Chaos, könnte also für eine gewichtete „Defokussierung“ geeignet sein.

4. Ein erstes Realisierungsmodell für ein semiprime factoring alternating gradient synchrotron

Im folgenden soll ein erstes Realisierungsmodell für ein semiprime factoring alternating gradient synchrotron vorgestellt werden, das auf den Rekursionsgleichungen (2) (für Beschleunigung und Fokussierung) und (9) (für gewichtete Defokussierung) basiert.

$$y(0), y(1) \in \mathbb{N} \tag{10}$$

$$y(t+1) = \begin{cases} \text{Floor} \left(y(t-1) \left((1-w) + w \frac{-1 + 3 \left(\frac{a}{y(t-1)y(t)} \right)^2}{3 - \left(\frac{a}{y(t-1)y(t)} \right)^2} \right) \right) & \text{für } t = 1, 3, 5, 7, \dots \\ & \text{Defokussierung} \end{cases} \tag{11}$$

$$\begin{cases} \left(\text{Ceiling} 1 \right) \left(1 + 3 \left(\frac{a}{y(t-1)y(t)} \right)^2 \right) & \text{für } t = 2, 4, 6, 8, \dots \\ & \text{Beschleunigung} \\ & \text{und Fokussierung} \end{cases}$$

$$a = \text{prime}(1) \cdot \text{prime}(2);$$

$$0 < w \ll 1 - \text{defocusing weight}$$

Beispiel 3

$$a = \text{prime}(1) \cdot \text{prime}(2) = 2619143683493;$$

$$\text{prime}(1) = \text{prime}[100000] = 1299709; \text{prime}(2) = \text{prime}[150000] = 2015177;$$

$$y(0) = \text{prime}(1) + 64500 = 1364209 \approx \text{prime}(1) \cdot (1 + 5/100);$$

$$y(1) = \text{prime}(2) - 1999573 = 15604 \approx \text{prime}(2) \cdot (1 - 99/100);$$

$$w = 9 \cdot 10^{-6}$$

Abarbeitungsprotokoll:

t	$y(t)$	t	$y(t)$
0	1364209	5924	1299721
1	15604	5925	2015159
2	1364159	5926	1299720
3	46804	5927	2015160
4	1364109	5928	1299719
5	140190	5929	2015162
6	1364059	5930	1299718
7	414686	5931	2015164
8	1364004	5932	1299717
9	1108329	5933	2015165
10	1302650	5934	1299716
11	1915567	5935	2015167
12	1302652	5936	1299715
13	2010568	5937	2015168
14	1302652	5938	1299714
15	2010625	5939	2015170
16	1302651	5940	1299713
17	2010626	5941	2015171
18	1302650	5942	1299712
19	2010628	5943	2015173
20	1302649	5944	1299711
21	2010629	5945	2015174
22	1302648	5946	1299710
23	2010631	5947	2015176
24	1302647	5948	1299709
25	2010632	5949	2015177
26	1302646	5950	1299709 = <i>prime</i> (1)
27	2010634	5951	2015177 = <i>prime</i> (2)
28	1302645	5952	1299709
29	2010636	5953	2015177
30	1302644	5954	1299709
31	2010637	5955	2015177

Das vorstehende Abarbeitungsprotokoll zeigt: Bei geeignet gewählten Parametern $y(0), y(1)$ und w ist **semiprime factoring prinzipiell möglich**.

Man erkennt dabei drei charakteristische Arbeitsbereiche:

1. Einschwingvorgang, hier von $t = 0$ bis $t = 15$:
starke Veränderungen von $y(t)$ hin zu zwei Werten, die sich im folgenden
2. Einschleichabschnitt, hier von $t = 16$ bis $t = 5947$:
 - 2-periodisch,
 - gegenläufig monoton und
 - nahezu linear
 nur noch wenig ändern, hier $\text{abs}(y(t) - y(t + 2)) \in \{1, 2\}$.
3. **Primzahlfangbereich**, hier ab $t = 5948$:

$(\forall t) y(t) = y(t + 2) \in \{\text{prime}(1), \text{prime}(2)\}$, d.h., das Modell liefert hier für den konstanten Input $a = \text{prime}(1) \cdot \text{prime}(2)$ die Outputsequenz $\dots, \text{prime}(1), \text{prime}(2), \text{prime}(1), \text{prime}(2), \dots$.

Beispiel 4

$a = \text{prime}(1) \cdot \text{prime}(2) = 11606868967288127684640649$;
 $\text{prime}(1) = \text{prime}[10 \cdot 10^{10}] = 2760727302517$;
 $\text{prime}(2) = \text{prime}[15 \cdot 10^{10}] = 4204279414597$;
 $y(0) = \text{prime}(1) + 6666 = 27607273309183$;
 $y(1) = \text{prime}(2) - 11111111111 = 4193168303486$;
 $w = 10^{-6}$

Abarbeitungsprotokoll:

t	$y(t)$	t	$y(t)$
0	2760727309183	72077	2760727302525
1	4193168303486	72078	4204279414585
2	2760727338561	72079	2760727302524
3	4204279340228	72080	4204279414587
4	2760727338561	72081	2760727302523
5	4204279359707	72082	4204279414588
6	2760727338560	72083	2760727302522
7	4204279359708	72084	4204279414590
8	2760727338559	72085	2760727302521
9	4204279359710	72086	4204279414591
10	2760727338558	72087	2760727302520
11	4204279359711	72088	4204279414593
12	2760727338557	72089	2760727302519
13	4204279359713	72090	4204279414594
14	2760727338556	72091	2760727302518
15	4204279359714	72092	4204279414596
16	2760727338555	72093	2760727302517
17	4204279359716	72994	4204279414597
18	2760727338554	72095	2760727302517 = $\text{prime}(1)$
19	4204279359717	72096	4204279414597 = $\text{prime}(2)$

Auch für größere Primzahlen $\text{prime}(1)$ und $\text{prime}(2)$ erhält man dasselbe Verhaltensmuster wie in Beispiel 3.

Beispiel 5

$a = \text{RSA-160} = 2152741102718889701896015201312825429257773588845675980170497676$
 $7781331452188591356730110597734910596024979071115852143020793146$
 $65202840140619946994927570407753$;
 $\text{prime}(1) = 45427892858481394071686190649738831656137145778469793250959984709250$
 004157335359 ;
 $\text{prime}(2) = 47388090603832016196633832303788951973268922921040957944741354648812$
 028493909367 [14] ;
 $y(0) = \text{prime}(1) + 33000$;
 $y(1) = \text{prime}(2) - 111111111111$;
 $w = 10^{-12}$

Der Primzahlfangbereich wird hier bei $t = 66003$ erreicht.

Weitere Experimente mit dem vorliegenden Modell zeigen:

- Das Erreichen des Primzahlfangbereiches hängt bei gegebenem a sehr empfindlich von den Parametern $y(0)$, $y(1)$ und w ab.
- In Abhängigkeit von der Rechengenauigkeit können Phantomfangbereiche mit $\dots, f(1), f(2), f(1), f(2), \dots$ aber $f(1) \cdot f(2) \neq a$ auftreten.
- Wird der Primzahlfangbereich verfehlt, mündet der Einschleichabschnitt schließlich in den terminalen Fangbereich mit der Outputsequenz $\dots, a, 1, a, 1, \dots$ ein.
- Verbesserungen und Erweiterungen des hier vorgestellten Primärmodells sind möglich und nötig. Das betrifft insbesondere die Verbreiterung und Symmetrierung des Initialisierungsintervalls $(y(0), y(1))$, für das über einen möglichst kurzen Einschleichabschnitt der Primzahlfangbereich erreicht wird, zu $(y(0), y(1)) = (\sqrt{a} + \Delta, \sqrt{a} - \Delta)$. Dazu kommen folgende Realisierungsmöglichkeiten in Betracht:
 - outputabhängige Steuerung von w im Einschwing- und Einschleichabschnitt,
 - datenabhängige Injektion in $y(t)$ gemäß Gl. (7),
 - Einbau eines outputgesteuerten Gewichtungsfaktors auch im Beschleunigungs- und Fokussierungsteil von Gl. (11),
 - Einbau von Oszillationselementen in Gl. (11) (Resonanz- und Mitnahmeeffekte, vgl. [10,11,12]),
 - Nutzung weiterer Problemlösungen aus der Synchrotrontechnologie (Stabilität, Fokussierung, Schwingungen, Tandembetrieb, ...) [8,9,13],
 - gekoppelter Betrieb von Produkt- und Summenzerleger (vgl. [7]).

5. Schlußbemerkung

Es ist im Umkehrschluß nahe liegend, die vorstehend betrachteten Rekursionsgleichungen (einschließlich Derivaten und Erweiterungen) auf ihre Eignung als Falltürfunktionen für public key Kryptosysteme zu untersuchen.

Der Autor dankt Dr. Dagmar Schönfeld für die technische Unterstützung und viele anregende Diskussionen bei der Abfassung dieses Moduls.

References

- [1] Crandall,R.E.; Pomerance,C.: *Prime Numbers. A Computational Perspective*. New York, Berlin, ...,Tokyo 2001
- [2] <http://mathworld.wolfram.com/RSAEncryption.html>
- [3] <http://mathworld.wolfram.com/Semiprime.html>
- [4] <http://www.crypto-world.com>
- [5] <http://www.rsasecurity.com>
- [6] <http://mathworld.wolfram.com/news/2005-05-10/rsa-200>
- [7] Stoschek,E.P.: *Abenteuer Algorithmus*. Teil 2, Kap. 4, Dresden 1997
- [8] Wilson,E.J.N.; Wilson,E.: *An Introduction to Particle Accelerators*. Oxford, New York 2001
- [9] Chao,A.W.; Tigner,M.: *Handbook of Accelerator Physics and Engineering*. Singapore 2002
- [10] <http://www.wissenschaft.de/wissen/news/257654.html>
- [11] Strogatz,S.H.; Abrams,D.M.; McRobie,A.; Eckhardt,B.; Ott,E.: *Crowd synchrony on the Millenium Bridge*. Nature 438, 43-44 (3 Nov. 2005)

- [12] Ito,T. ; Ito,K. : *Nonlinear dynamics of homeothermic temperature control in skunk cabbage, *Symplocarpus foetidus**. Phys. Rev. E 72, 051909 (2005)
- [13] Plettner,T. et al. : *Visible-Laser Acceleration of Relativistic Electrons in a Semi-Infinite Vacuum*. Phys. Rev. Lett. 95, 134801 (2005)
- [14] <http://www.rsasecurity.com/rsalabs/node.asp?id=2097>

Are you looking for suggestions and ideas on algorithms, mathematical, and computer-aided graphic design (textile, ceramics, book illustration, calligraphy, logo, advertising etc.)?

Visit <http://www.DresdenAlgorithmicsChannel.de> !

e.p.stoschek	17.11.2005
stoschek@tcs.inf.tu-dresden.de	