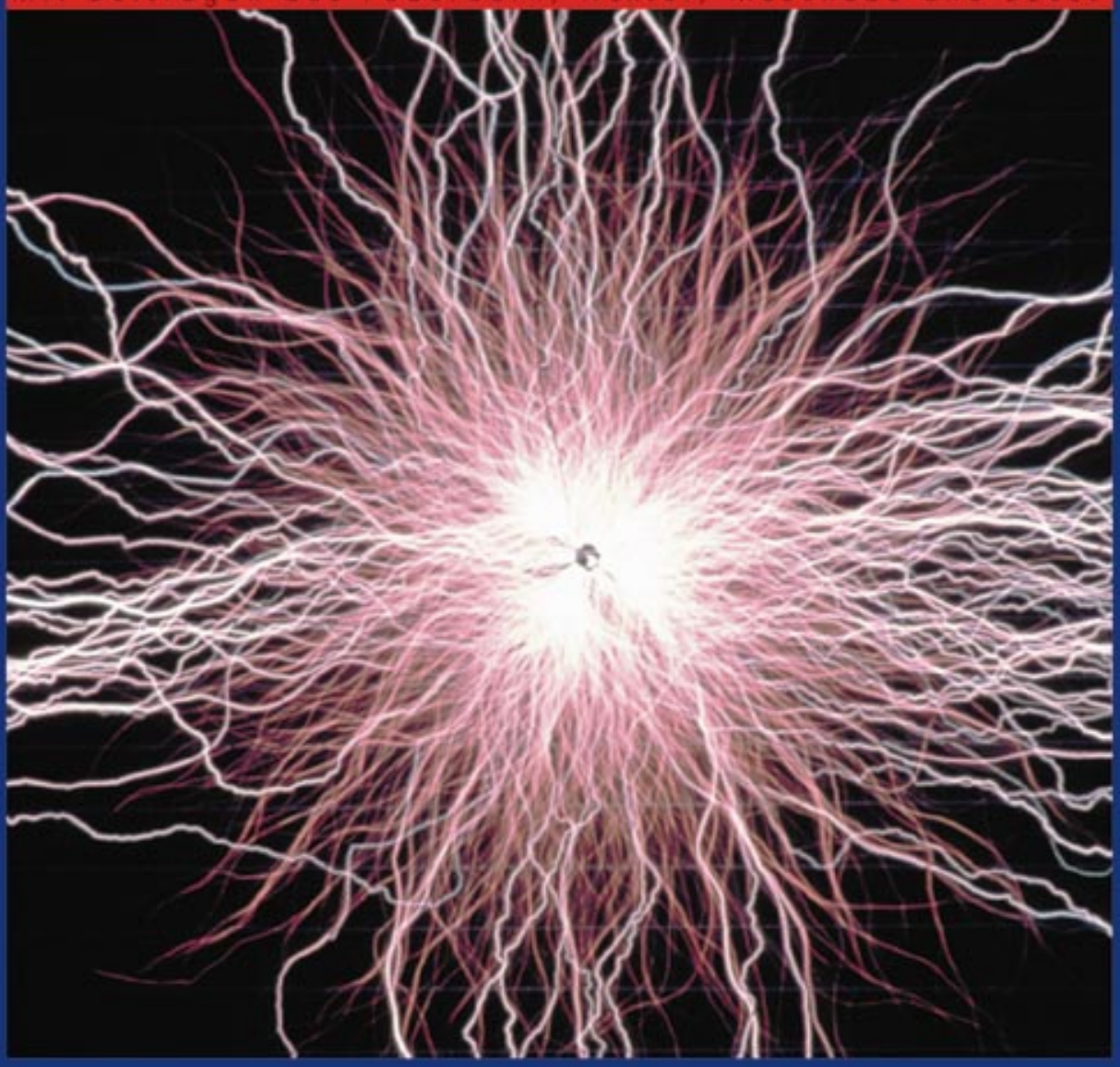
FORSCHUNGS FORUM

PADERBORN

Paderborner Universitätsmagazin 1-1998

Mit Beiträgen aus Paderborn, Höxter, Meschede und Soest



- **Kryptographie**
  Entschlüsseln Sie das Geheimnis Paderborns!
- **Digitale Röntgenbilder für die Medizin**
- **Überholen auf der Datenautobahn**
- **Schulen in der Informationsgesellschaft**
- **Inspektionstechnologie für das Kanalnetz**

Algebra für Spione, Datenschützer und das Internet

Kryptographie und endliche Körper

Moderne algebraische und zahlentheoretische Methoden haben die Kryptographie revolutioniert - mit Anwendungen bei vielen sicheren Formen der Datenübertragung (ec-Karten etc.). Dieser Artikel gibt eine kurze Übersicht und beschreibt ein spezielles algorithmisches Problem.

Das Verschlüsseln geheimer Nachrichten gehört zum Handwerkszeug des zweitältesten Gewerbes der Welt: der Spionage. Julius Caesar kodierte seine Nachrichten, indem er jeden Buchstaben um eine feste Größe verschob. Dabei identifiziert man die 26 Buchstaben unseres Alphabets mit 26 Zahlen, z.B.

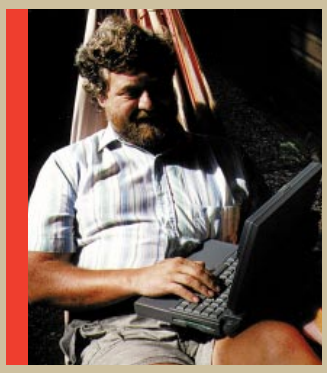
A	B	C	D	E	...	X	Y	Z
0	1	2	3	4	...	23	24	25

Nun legt man einen geheimen *Schlüssel* k fest, z.B. $k = 10$, und die Kodierung $\kappa(\mathsf{a})$ eines Buchstaben a ist einfach $\kappa(\mathsf{a}) = \mathsf{a} + k$. Wörter oder längere Texte werden Buchstabe für Buchstabe kodiert, und zum Dekodieren eines Buchstabens b bildet man $\delta(b) = b - k$. Als Beispiel:

$$\begin{array}{lcccl} \text{CAESAR} & \leftrightarrow & (2,0,4,18,0,17) & = & \mathsf{x} = \delta(\mathsf{y}) \\ \mathsf{y} = \kappa(\mathsf{x}) & = & (12,10,14,2,10,1) & \leftrightarrow & \text{MKOCKB} \end{array}$$

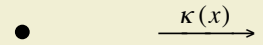
Foto: Braun, HNF

Abb. 1: Die deutsche ENIGMA-Chiffriermaschine aus dem 2. Weltkrieg. Dieses Exemplar steht im Heinz Nixdorf MuseumsForum, Paderborn.



Joachim von zur Gathen,
Fachbereich 17/Mathematik - Informatik, ist (nach 14 Jahren an der University of Toronto) seit 1994 Professor für Algorithmische Mathematik an der Universität Paderborn. Er beschäftigt sich mit Komplexitätstheorie und Computeralgebra.

Der Absender schickt also *MKODKC*, und der Empfänger dekodiert das zu *CAESAR*.



Sender: $x \mapsto \kappa(x) = y$ Empfänger: $y \mapsto \varepsilon(y) = x$

Beim Kodieren kommt A hinter Z, so daß $\kappa(18) = 28 = 3 \leftrightarrow D$.

Diese Verschlüsselungsmethode ist äußerst *einfach* und *ebenso unsicher*. Den geheimen Schlüssel $k = 10$ kann man leicht durch Raten herausfinden (es gibt ja nur 26 Möglichkeiten), und manche Verallgemeinerung dieses klassischen Verfahrens fällt einer Frequenzanalyse der Buchstaben zum Opfer - vorausgesetzt, man kennt die Sprache. (Das US-Militär hat im 2. Weltkrieg die Indianersprache Navajo benutzt.)
Die raffiniertesten „klassischen" Methoden basieren auf solchen Permutationen von Buchstabengruppen und sind nicht leicht zu brechen. Die deutsche Wehrmacht und Marine benutzten im 2. Weltkrieg das *Enigma-System* (Abbildung 1). Einer Gruppe englischer Wissenschaftler in Bletchley Park unter Leitung von Alan Turing, dem Begründer der theoretischen Informatik, gelang es mit dem Einsatz ihres *Colossus-Computers*, dieses System (zeitweise) zu brechen; dies war für den Ausgang der U-Boot-Schlacht im Nordatlantik entscheidend. Der Roman *Enigma* von Robert Harris schildert eindringlich die Stimmung in diesem Mathematikercamp.

Kodiertafeln und visuelle Kryptographie

Eine weitere klassische Methode ist der Gebrauch von Kodiertafeln, die von Vernam 1926 vorgeschlagen wurde. Diese bestehen aus langen Zahlenreihen, möglichst zufällig ausgewählt. Abbildung 2 zeigt eine solche Tafel, wie sie eine enttarnte DDR-Spionin in einem Kleiderbügel versteckt hatte. Zur Übermittlung

Foto: Fechner/laife

Abb. 2: In einem Kleiderbügel versteckte MfS-Kodiertafel.

wandelt die *Kundschafterin des Friedens* den eigentlichen Text nach einem festgelegten, einfachen Verfahren in Blöcke von fünfstelligen Zahlen und addiert dann blockweise die Zahlen der Kodiertafel dazu. Dabei bleibt alles fünfstellig, d.h., es wird modulo 100000 gerechnet. Wenn also etwa die Nachricht 62103 ist und dazu der Block 55126 aus Abbildung 2 gehört, so schickt sie 12229. Ihr Agentenführer an der Normannenstraße subtrahiert davon einfach den Block aus der Tafel - von der er auch eine Kopie hat - und schon ist wieder ein Quentchen mehr über den kapitalistischen Gegner bekannt. (Das Urteil der Historie über die erfolgreiche DDR-Spionage ist wohlbekannt ...)

Wenn die Kodiertafel nur einmal benutzt wird, so ist dieses Verfahren absolut *sicher*, aber sehr *umständlich zu benutzen*, da die Agentin mit genügend Tafeln versorgt werden muß, um die gesamte Länge aller ihrer Nachrichten abzudecken.

Die *visuelle Kryptographie* von Naor und Shamir liefert eine graphische Illustration der Kodiertafeln. Dabei wird die Bildfläche in *Großpixel* unterteilt, von denen jedes aus 2x2 Pixeln (Bildelementen) besteht. Die visuelle Kodiertafel besteht aus Großpixeln mit je zwei weißen und zwei schwarzen Pixeln (Abbildung 3a). Die sechs Möglichkeiten werden dabei zufällig ausgesucht. Das (z.B. per Fax) übertragene Bild wird nach Abbildung 3b erzeugt; es sieht ebenfalls zufällig aus. Wenn die beiden Figuren übereinandergelegt werden, so ist das „geheime" Bild sichtbar. **Der Leser kann das auf Seite 13 mit Hilfe der am Ende der Zeitschrift beigefügten Folie ausprobieren.**

Algebraische Geheimnisse

Als Kryptosystem ist Caesars Methode nicht zu empfehlen, aber sie ist ein einfaches Beispiel für eine ganz wichtige Idee: die Benutzung der **Algebra**. Der erste Schritt hierbei ist die Identifizierung des Alphabets mit einer algebraischen Struktur: dem Ring Z_{26} der Zahlen 0, 1, ..., 25, wobei „modulo 26" gerechnet wird, so daß $18 + 10 \equiv 2 \bmod 26$. Der zweite Schritt ist die Wahl einer arithmetischen Funktion κ zum Kodieren und ihrer Inversen δ zum Dekodieren. Im Beispiel ist $\kappa(a) = a + 10$ und $\delta(a) = a - 10 = a + 16$. Dieses Prinzip ist lange wohlbekannt. So sagte der Algebraiker Abraham Adrian Albert vor einer Versammlung der American Mathematical Society 1941: „It would not be an exaggeration to state that *abstract* cryptography is *identical* with abstract mathematics." („Es wäre keine Übertreibung zu behaupten, daß *abstrakte* Kryptographie *identisch* mit abstrakter Mathematik ist.")

Die zweite Idee ist neueren Datums und hat die Kryptographie revolutioniert. Diffie und Hellman haben nämlich 1976 vorgeschlagen, Kryptographie „mit öffentlichem Schlüssel" zu machen. Wenn also Bob eine Nachricht an Alice schicken will - so heißen die Figuren üblicherweise in der Literatur - so stellt Alice einen geheimen Schlüssel *S* und einen öffentlichen Schlüs-

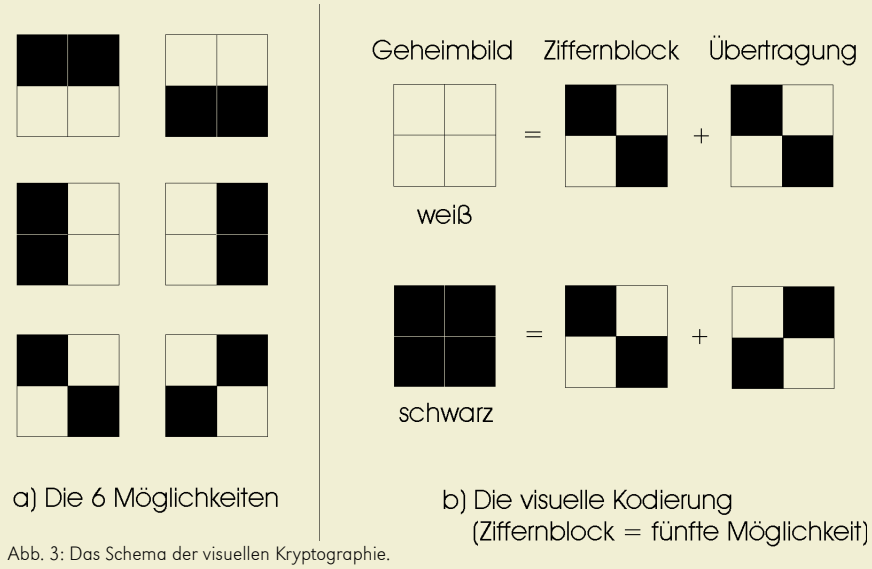
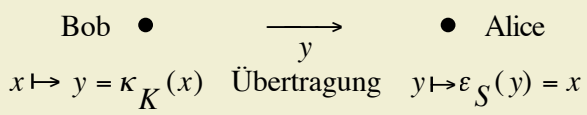


Abb. 3: Das Schema der visuellen Kryptographie.

sel K her. Jedermann - also auch Bob - kennt K und die entsprechende Verschlüsselungsfunktion κ_K. Nur Alice kennt die Entschlüsselungsfunktion ε_S, mit der Eigenschaft, daß

$$\varepsilon_S(\kappa_K(x)) = x.$$

Dabei sollen die folgenden Eigenschaften gelten:



- κ_K ist leicht auszuführen,
- ε_S ist leicht auszuführen, wenn man S kennt,
- ε_S ist schwierig auszuführen, wenn man S nicht kennt.

Die zugrundeliegende Abbildung ist eine *Falltürfunktion*: man kommt leicht rein, aber nur schwer wieder raus, außer wenn man den geheimen Schlüssel hat.
Hier spielt nun die *Komplexitätstheorie*, ein erfolgreiches und vitales Teilgebiet der theoretischen Informatik, eine entscheidende Rolle. Sie liefert nämlich - im Prinzip jedenfalls - Werkzeuge, um die zunächst vagen Begriffe „*leicht*“ und „schwierig“ präzise zu fassen. Eine Aufgabe heißt nämlich leicht, wenn es eine Methode gibt, die sie in polynomialer Zeit löst, d.h. mit Aufwand etwa n^2, wobei n die Länge der Eingabe in geeigneter Weise beschreibt, und statt 2 irgendeine feste Zahl genommen werden kann. Dabei sind auch Zufallswahlen innerhalb des Algorithmus zugelassen, vorausgesetzt, das Ergebnis ist mit überwältigender Wahrscheinlichkeit richtig. Was nicht leicht ist, heißt *schwierig*.
Seit der Pionierleistung von Diffie und Hellman hat sich die moderne Kryptographie rasant entwickelt. Sie hat längst die Schattenwelt der Schlapphüte verlassen und wird heute überall dort in der elektronischen Datenübertragung eingesetzt, wo Daten geheim bleiben sollen: Geldtransfers quer über den Globus, ec-Karten, Bestellungen im Internet etc. Die bekannteste Methode ist das RSA-Protokoll von Rivest, Shamir und Adleman. Wie mehrere solche Verfahren beruht es auf raffinierten Anwendungen der Zahlentheorie, so daß David Hilberts Erkenntnis von 1930 heute nicht mehr gilt: „Die reine Zahlentheorie ist dasjenige Gebiet der Mathematik, das bisher noch nie Anwendung gefunden hat“.

Das ElGamal-Kryptosystem

Im Folgenden sei ein modernes Kryptosystem beschrieben - das *ElGamal-System* - zu dessen Algorithmik wir Beiträge geliefert haben. Grundlage sind die *endlichen Körper*, wie für viele Anwendungen der Algebra, z.B. beim Versuchsentwurf, dem Korrigieren von Nachrichten über verrauschte Kanäle, und den endlichen Geometrien. Ein solcher endlicher Körper besteht aus endlich vielen Elementen, sagen wir q Elementen, mit denen man wie üblich rechnen kann: addieren, subtrahieren, multiplizieren, dividieren (außer durch Null). Das einfachste Beispiel ist $Z_2 = \{0,1\}$, mit $1 + 1 = 0$, d.h. Rechnen modulo 2. Allgemein existiert so etwas genau dann, wenn q eine Potenz einer Primzahl ist, z.B. für alle q mit $2 \le q \le 9$ außer für $q = 6$. Wir bezeichnen das dann mit F_q.
In F_q gibt es stets ein *erzeugendes Element* g (sogar viele solche), dessen Potenzen g, g^2, g^3,... sämtliche Elemente (außer Null) des Körpers bilden. Alice wählt nun solche q, g und außerdem eine ganze Zahl b mit $2 \le b < q$, berechnet $w = g^b$ und veröffentlicht ihren öffentlichen Schlüssel $K = (q, g, w)$; ihr geheimer Schlüssel ist $S = b$. Wenn Bob ihr eine Nachricht x schicken will (wir können annehmen, daß $x \in F_q$), so wählt er eine zufällige Zahl $k < q$, berechnet $u = g^k$ und $v = xw^k$ und schickt $\kappa_K(x) = (u,v)$ an Alice. Sie kann dann die Nachricht leicht ausrechnen mittels $x = v / u^b$.

Die Crux dieses Verfahrens ist, daß das Potenzieren in endlichen Körpern eine Falltürfunktion ist. Das schnelle Berechnen einer Potenz a^e ist für $e = 13$ in Abbildung 4 gezeigt. Im allgemeinen geht das mit höchstens $2\log_2 e$ Multiplikationen - das ist „leicht”. Die Theorie der *Additionsketten* liefert raffinierte Verbesserungen dieser Methode, bis zu wenig mehr als $\log_2 e$ Multiplikationen. Weil eine Multiplikation den Grad höchstens verdoppeln kann, braucht man auch mindestens $\lceil \log_2 e \rceil$ Multiplikationen.

$$e = 13 = 1\cdot 2^3 + 1\cdot 2^2 + 0\cdot 2^1 + 1\cdot 2^0 \text{ in Binärdarstellung}$$
$$r_1 = a^2 \qquad r_2 = r_1^2 (= a^4) \qquad r_3 = r_2^2 (= a^8)$$
$$r_4 = a \cdot r_2 (= a^5) \qquad r_5 = r_4 \cdot r_3 (= a^{13})$$

Abb. 4: Berechnen von a^{13} mittels wiederholtem Quadrieren.

Die Umkehrabbildung ist der *diskrete Logarithmus*: Zu gegebenen $a \in F_q$ soll man ein e berechnen mit $g^e = a$, so daß $e = \log_g a$. Dies ist ein wohlstudiertes Problem, aber die schnellsten bekannten (und mathematisch tiefliegenden) Algorithmen brauchen exponentielle Zeit, ungefähr $2^{n^{1/3}}$ Operationen - genauer: $\exp(O((n\log^2 n)^{1/3}))$ - für ein n-stelliges q; das zählt als „schwierig”. Das “O“ steht für eine nicht näher spezifizierte Konstante. Genau genommen muß der Angreifer des ElGamal-Systems ein etwas anderes Problem lösen: aus g^k und g^b soll g^{kb} berechnet werden. Aber alle bekannten Algorithmen hierfür benutzen den diskreten Logarithmus. Somit sind die drei Eigenschaften einer Falltürfunktion vermutlich gegeben.
Die Firma Newbridge Microsystems aus Ontario, Kanada, hat in Zusammenarbeit mit Kryptologen der University of Waterloo einen erfolgreichen Kryptoprozessor auf Basis dieses Verfahrens entwickelt. Der benutzte Körper $F_{2^{593}}$ hat 2^{593} Elemente; eine Nachfolgeversion benutzt $F_{2^{1013}}$. Der Prozessor ist in Abbildung 5 zu sehen, und die Kosten einiger Anweisungen in Abbildung 6. Im von mir rot umrandeten Teil sind drei Potenzierroutinen aufgeführt, mit drastisch unterschiedlichen Laufzeiten. Welche davon wird der Kunde wohl wählen? Natürlich die mit der kürzesten Rechenzeit und somit dem höchsten Durchsatz. Niemand sagt ja dem Kunden, daß damit die allgemeinen Überlegungen zur Sicherheit des Systems ihre Gültigkeit verlieren.

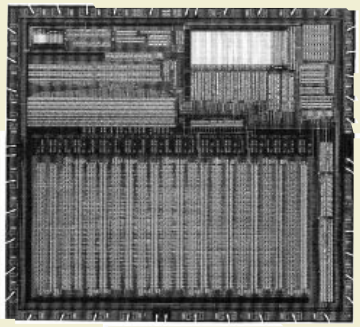


Abb. 5: Der Newbridge Kryptoprozessor.

Instruction* OP Code		Operation	Register A	Register B	D	R	S	DEP cycles
Class 2								
INVA	D0	Compute the inverse of A (Note 1)	A^{-1}	U				50000
MULT	A0	A gets A*B	A*B					1300
EXP1	D0	Full exponentiation	$(A)^{D}$					up to 10^6
EXP2	C4	Fast exponentiation (Note 3)	$(A)^{Rm'}$	$(A)^{Rm'}$				up to 40000
EXP3	CC	EXP2						up to 80000
		PERMR						
		CPA2B						
		SWAPRS						
		EXP2						
		PERMR						
		SWAPRS (Note 2)	$(A')^{f(R'',S'')}$	$(A')^{f(R'',S'')}$		R'	S'	

Abb. 6: Die Zeiten für einige Instruktionen in Data Encryption Processor (DEP) Zyklen. Für uns relevant sind die umrandeten Potenzieranweisungen.

Schnelles Potenzieren

Das allgemeine Forschungsgebiet meiner Arbeitsgruppe in Paderborn sind der Entwurf und die Implementierung schneller Algorithmen für grundlegende Aufgaben der Computeralgebra. Innerhalb des SFB „Massive Parallelität“ wird vor allem an Methoden für parallele Rechner gearbeitet; dabei hat sich eine erfolgreiche Zusammenarbeit mit der theoretischen Informatik (Meyer auf der Heide, Monien) und den Entwicklern des MuPAD-Computeralgebrasystems (Fuchssteiner) ergeben.

Unser Ziel war es nun, einen schnelleren Algorithmus für die Grundaufgabe des Potenzierens in endlichen Körpern zu finden. Die Lösung hat mehrere Komponenten, die hier nur oberflächlich beschrieben werden können. Die erste ist die *schnelle Multiplikation*. Wenn man die übliche Formel für das Produkt zweier n-stelliger Zahlen (oder zweier Polynome vom Grad n) nimmt, so liefert das einen Algorithmus mit ungefähr $2n^2$ Operationen. Diese Kosten können verringert werden; der Weltrekord ist seit über einem Vierteljahrhundert der Algorithmus von Schönhage und Strassen, mit nur $O(n \log n \log\log n)$ Operationen. Er beruht auf einer geschickten Anwendung der schnellen Fouriertransformation; es ist ein ungelöstes Forschungsproblem, ob er verbes-

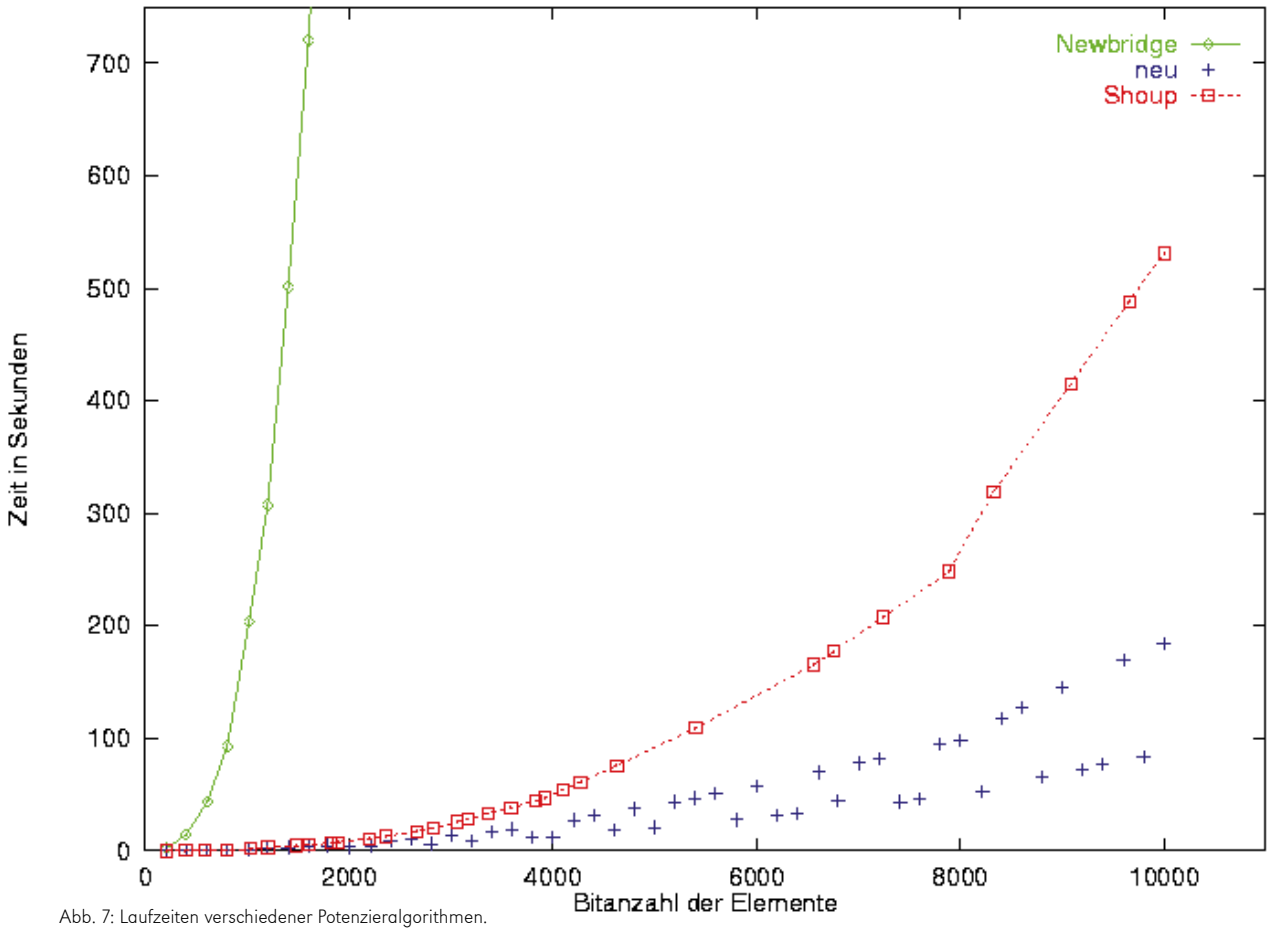


Abb. 7: Laufzeiten verschiedener Potenzieralgorithmen.

sert werden kann (z.B. um den Faktor loglog$n$).
Die zweite Komponente sind *Gauß-Perioden*, eine Methode, mit der Carl Friedrich Gauß 1798 das seit der Antike offene Problem löste, regelmäßige n-Ecke mit Zirkel und Lineal zu konstruieren. Seit Gauß weiß man, daß dies genau dann geht, wenn n ein Produkt von Zweierpotenzen und Primzahlen von der Form $2^k + 1$ ist; diese dürfen n nur einmal teilen, z.B.:

3, 4, 5, 6, 8, 10, 12, 15, 17.

Diese Gauß-Perioden wurden schon im Newbridge-Kryptoprozessor eingesetzt. Aber erst 1994 haben Gao, von zur Gathen und Panario gezeigt, wie man Gauß-Perioden mit schneller Multiplikation zusammen verwenden kann. Dies drückt die Potenzierkosten in F_q, wo q eine n-stellige Zahl ist, von $O(n^3)$ auf $O(n^2 \log n \log\log n)$ Operationen.
Den wichtigen Schritt in die Praxis hat Michael Nöcker in seiner Diplomarbeit getan. Neben Untersuchungen zu Additionsketten hat er die schnellen Potenzieralgorithmen implementiert. Abbildung 7 zeigt einige Messungen; hierbei ist „Newbridge“ die im Kryptoprozessor verwendete Methode, „Shoup“ ein neuer Zugang von Victor Shoup und „neu“ unser Algorithmus. Die neue Methode wird bereits in Waterloo eingesetzt.
Diese Gauß-Perioden existieren nicht in allen Fällen, sondern z.B. in F_{2^n} nur für etwa 26% aller $n \le 1000$. (Dies bezieht sich auf *optimale Gauß-Perioden*, auf deren Definition ich hier verzichte.) Für kryptographische Zwecke ist dies ausreichend, aber trotzdem sucht man nach einer breiter anwendbaren Methode. In ihrer Diplomarbeit hat Sandra Schlink dies erreicht; dabei muß ein gewisser Parameter nur noch quadratfrei (ohne mehrfache Faktoren) sein, der im klassischen Zugang eine Primzahl sein mußte. In Zusammenarbeit mit Amin Shokrollahi während meines Forschungsfreisemesters in Berkeley gelang es, auch diese letzte Einschränkung zu beseitigen (Feisel, von zur Gathen, Shokrollahi; dazwischen feierten wir Sandras Hochzeit mit Mirko Feisel). Der Fortschritt sieht im Beispiel so aus:

Parameterwert:	41	39	45
klassisch	ja	nein	nein
Schlink	ja	ja	nein
Feisel, von zur Gathen, Shokrollahi	ja	ja	ja

Als nächste Forschungsaufgabe innerhalb der beantragten Weiterführung des Sonderforschungsbereichs haben wir uns gestellt, diese Methoden - von der schnellen Arithmetik bis zu den Gauß-Perioden - auf massiv parallele Rechner zu übertragen.

Literatur

W. DIFFIE AND M. E. HELLMAN, New directions in cryptography. IEEE Trans. Inform. Theory 22 (1976), 644-654.
T. ELGAMAL, A public key cryptosystem and a signature scheme based on discrete logarithms. IEEE Transactions on information theory IT-31(4) (1985), 469-472.
S. FEISEL, J. VON ZUR GATHEN, AND M. A. SHOKROLLAHI, Normal bases via general Gauß periods. Erscheint in Mathematics of Computation, 1998.
S. GAO, J. VON ZUR GATHEN, AND D. PANARIO, Gauß periods and efficient arithmetic in finite fields. Extended abstract in Proc. LATIN '95, Lecture Notes in Computer Science 911 (1995), 311-322.
J. VON ZUR GATHEN AND M. NÖCKER, Exponentiation in finite fields: Theory and practice. In Proc. 12th Symp. Applied Algebra, Algebraic Algorithms and Error-Correcting Codes, AAECC-12, Lecture Notes in Computer Science 1255 (1997), 88-113.
J. VON ZUR GATHEN AND S. SCHLINK, Normal bases via general Gauss periods. Reihe Informatik, tr-ri-96-177, Universität-Gesamthochschule Paderborn, 1996.
R. HARRIS, Enigma, Roman. Heyne Verlag, 1996.
M. NAOR AND A. SHAMIR, Visual Cryptography, in EUROCRYPT '94, Lecture Notes in Computer Science 950 (1994), 1-12.
M. NÖCKER, Exponentiation in finite fields; theory and practice, Diplomarbeit. 1996.
R. L. RIVEST, A. SHAMIR, AND L. M. ADLEMAN, A method for obtaining digital signatures and public-key cryptosystems. Comm. ACM 21 (1978), 120-126.
T. ROSATI, A high speed data encryption processor for public-key cryptography. In Proc. of IEEE Custom Integrated Circuits Conference, San Diego, 1989, 12.3.1 - 12.3.5.
S. SCHLINK, Normalbasen mit Hilfe von verallgemeinerten Gauß-Perioden, Diplomarbeit, 1996.
A. SCHÖNHAGE AND V. STRASSEN, Schnelle Multiplikation großer Zahlen. Computing 7 (1971), 281-292.
G. S. VERNAM: Cipher printing telegraph systems for secure wire and radio telegraphic communications, J. Amer. Inst. Elect. Eng. XLV (1926), 109-115.