

Oktober 2019

# Computeralgebra Rundbrief

> Ausgabe 65

- ▶ Tagung der Fachgruppe 2019
- ▶ Isogeny-based Cryptography
- ▶ Towards Post-Quantum Symmetric Cryptography
- ▶ Eine Gerade dreht sich um eine andere: Die Kühlturmfläche



Teachers Teaching with Technology™

## Netzwerk | Fortbildungen | Materialien

Das T³ Lehrerfortbildungsnetzwerk richtet sich an Sie, an Lehrerinnen und Lehrer, die sich zum sinnvollen Einsatz digitaler Werkzeuge im MINT-Unterricht austauschen und weiterentwickeln wollen.

Machen Sie mit! Nehmen Sie an unseren Fortbildungen teil, stöbern Sie in unseren Unterrichtsmaterialien, tauschen Sie mit uns Ihre Erfahrungen aus.



Informieren Sie sich.  
Nehmen Sie Kontakt zu uns auf!

[www.t3deutschland.de](http://www.t3deutschland.de)  
[info@t3deutschland.de](mailto:info@t3deutschland.de)



## Inhaltsverzeichnis

<b>Inhalt</b>	3
<b>Impressum</b>	4
<b>Mitteilungen der Sprecher</b>	5
<b>Kandidatinnen und Kandidaten für die Fachgruppenleitung</b>	6
<b>Tagungen der Fachgruppe</b>	8
<i>Tagung der Fachgruppe Computeralgebra, Kassel 2019</i>	8
<i>Minisymposium Computeralgebra, DMV-Tagung</i>	10
<b>Themen und Anwendungen</b>	12
<i>Isogeny-based Cryptography</i> (C. Martindale, L. Panny)	12
<i>Towards Post-Quantum Secure Symmetric Cryptography</i> (X. Bogomolec, J. G. Underhill, S. A. Kovac)	18
<b>Computeralgebra in der Schule</b>	26
<i>Eine Gerade dreht sich um eine andere: Die Kühlturmfläche</i> (J. Meyer)	26
<b>Berichte über Arbeitsgruppen</b>	30
<i>SFB/TRR 195 Symbolic Tools in Mathematics and their Application (Part 4/5)</i>	30
<i>Computeralgebra in Zahlentheorie und Arithmetischer Geometrie (Würzburg)</i>	31
<b>Publikationen über Computeralgebra</b>	32
<b>Promotionen in der Computeralgebra</b>	33
<b>Berufungen</b>	33
<b>Berichte von Konferenzen</b>	34
<b>Hinweise auf Konferenzen</b>	36
<b>Fachgruppenleitung Computeralgebra 2017–2020</b>	39

## Impressum

Der Computeralgebra-Rundbrief wird herausgegeben von der Fachgruppe Computeralgebra der GI in Kooperation mit der DMV und der GAMM (verantwortlicher Redakteur: Dr. Fabian Reimers [car@mathematik.de](mailto:car@mathematik.de))

Der Computeralgebra-Rundbrief erscheint halbjährlich, Redaktionsschluss 15.02. und 15.09. ISSN 0933-5994. Mitglieder der Fachgruppe Computeralgebra erhalten je ein Exemplar dieses Rundbriefs im Rahmen ihrer Mitgliedschaft. Fachgruppe Computeralgebra im Internet: <http://www.fachgruppe-computeralgebra.de>.

Konferenzankündigungen, Mitteilungen, einzurichtende Links, Manuskripte und Anzeigenwünsche bitte an den verantwortlichen Redakteur.

**GI** (Gesellschaft für  
Informatik e.V.)  
Wissenschaftszentrum  
Ahrstr. 45  
53175 Bonn  
Telefon 0228-302-145  
Telefax 0228-302-167  
[gs@gi-ev.de](mailto:gs@gi-ev.de)  
<http://www.gi-ev.de>



**DMV** (Deutsche Mathematiker-  
Vereinigung e.V.)  
Mohrenstraße 39  
10117 Berlin  
Telefon 030-20377-306  
Telefax 030-20377-307  
[dmv@wias-berlin.de](mailto:dmv@wias-berlin.de)  
<http://www.dmv.mathematik.de>



**GAMM** (Gesellschaft für Angewandte  
Mathematik und Mechanik e.V.)  
Technische Universität Dresden  
Institut für Statik und Dynamik der  
Tragwerke  
01062 Dresden  
Telefon 0351-463-33448  
Telefax 0351-463-37086  
[GAMM@mailbox.tu-dresden.de](mailto:GAMM@mailbox.tu-dresden.de)  
<http://www.gamm-ev.de>



Liebe Mitglieder der Fachgruppe Computeralgebra,

seit dem letzten Rundbrief hat sich einiges getan. An erster Stelle sei hier die in jedem zweiten Jahr stattfindende Tagung der Fachgruppe erwähnt, die auch diesmal wieder am bewährten Ort in Kassel ausgerichtet wurde. Gerade durch ihre Regelmäßigkeit und die Zielgruppe des Nachwuchses nimmt sie einen besonderen Stellenwert unter unseren Aktivitäten ein. Ein ausführlicher Bericht zu der Tagung findet sich auf Seite 8. Doch diese siebte erfolgreiche Tagung, die Wolfram Koepf mit seinem Team für die Fachgruppe organisiert hat, wird leider auch die letzte unter seiner Federführung sein. Er ist dieses Jahr in den Ruhestand getreten. Wir möchten ihm an dieser Stelle für seine unermüdliche Arbeit für die Fachgruppe und auch seinem Team in Kassel für die allzeit verlässliche und professionelle Unterstützung danken. Die nächste Tagung der Fachgruppe im Jahr 2021 wird an einem neuen Ort stattfinden, am Max-Planck-Institut für Physik in München unter der Federführung von Thomas Hahn und Gregor Kemper.

Mit Wolfram Koepfs Weggang aus Kassel verbinden sich aber auch organisatorische Änderungen bei der Fachgruppe. Kaum bemerkt von den Nutzern lagen die Webseiten und die Mailinglisten der Fachgruppe bisher auf einem Server der Universität Kassel, der inzwischen veraltet ist und demnächst abgeschaltet wird. Die Fachgruppenleitung hat auf ihrer Sitzung am 30.9. in Aachen Details für die Migration dieser Dienste zu anderen Anbietern besprochen, die in den nächsten Wochen erfolgen wird. Wir hoffen, dass der Serverwechsel genauso unbemerkt von staten gehen wird wie der bisherige Betrieb. Auch hier nochmals großer Dank an Wolfram Koepf und an seinen Systemverwalter.

Doch die Tagung in Kassel war nicht die einzige Aktivität der Fachgruppe in diesem Jahr. Auf der Jahrestagung der DMV in Karlsruhe fand unter der Federführung von vier Mitgliedern der Fachgruppenleitung ein Minisymposium zur Computeralgebra statt, über das ebenfalls in der Rubrik "Tagungen der Fachgruppe" ab Seite 10 berichtet wird. Schon jetzt steht außerdem fest, dass auf der Jahrestagung der GAMM im Frühjahr 2020 ebenfalls ein Minisymposium zur Computeralgebra stattfinden wird, zu dem Details in den Konferenzhinweisen am Ende des Heftes zu finden sind.

Das vielleicht wichtigste Thema dieser Mitteilungen ist jedoch die schon im letzten Rundbrief angekündigte **Wahl der Fachgruppenleitung**. Es werden insgesamt neun Mitglieder der Fachgruppenleitung (mit einer Amtszeit von drei Jahren ab Frühjahr 2020) gewählt, daher hat jedes Mitglied der Fachgruppe bis zu neun Stimmen, wobei keine Häufung der Stimmen möglich ist. Abgesehen von den gewählten Mitgliedern besteht die Fachgruppenleitung aus je einer Vertreterin oder einem Vertreter der DMV, GI und GAMM, die von diesen Organisationen entsandt werden, sowie bis zu drei durch die Fachgruppenleitung berufenen Fachexpertinnen und -experten. Die Wahlleitung wird dieses Mal gebildet von Wolfram Koepf und Eva Zerz, die als Vertreter der DMV bzw. GAMM selbst nicht zur Wahl stehen. Ab Seite 6 finden Sie die Kurzvorstellungen der Kandidatinnen und Kandidaten.

Als Neuerung findet die Wahl erstmals nicht nur in Papierform, sondern vorrangig elektronisch statt. Dies soll das Wählen vereinfachen und damit die Beteiligung an der Wahl fördern. Sofern uns Ihre E-Mail-Adresse vorliegt, sollten Sie auf diesem Wege bereits eine Wahlnachricht erhalten haben. Alternativ besteht weiter die Möglichkeit, die klassische Briefwahl zu benutzen. Die entsprechenden Unterlagen liegen diesem Rundbrief bei. Falls—aus welchen Gründen auch immer—bei der Wahlleitung eine elektronische und eine schriftliche Stimmabgabe eingehen, so hat die elektronische Vorrang. Sowohl elektronisch als auch in Papierform abgegebene Stimmen müssen bis zum

**Samstag, den 30. November 2019**

die Wahlleitung erreicht haben.

Ansichts so viel Informationen in eigener Sache darf aber auch die Computeralgebra nicht zu kurz kommen. Beide Artikel der Rubrik "Themen und Anwendungen" des vorliegenden Rundbriefs beschäftigen sich mit verschiedenen Aspekten des Themas Postquanten-Kryptographie, allerdings aus ganz unterschiedlichen Blickwinkeln. Dies war nicht von langer Hand geplant sondern hat sich – nicht zuletzt durch die große Aktualität dieses Themas – zufällig ergeben. Ein Artikel aus dem Bereich "Computeralgebra in der Schule" beschäftigt sich mit Flächen, die durch das Bewegen einer Gerade entstehen, und rundet diese Ausgabe ab.

Damit bleibt uns nur noch, Ihnen eine angenehme und anregende Lektüre dieses Hefts zu wünschen.

Gregor Kemper

Anne Frühbis-Krüger



---

## Kandidatinnen und Kandidaten für die Fachgruppenleitung

---



**Dipl. Math. Xenia Bogomolec**, selbstständige IT-Fachkraft in verschiedenen industriellen Bereichen, hauptsächlich IT-Security. Diplom in algorithmischer kommutativer Algebra. Breites Netzwerk zwischen Industrie und Wissenschaft. Seit 2018 als Fachexpertin Industrie in der Fachgruppenleitung.

<https://quant-x-sec.com>



**Prof. Dr. Michael Cuntz**, Professor für Diskrete Mathematik an der Leibniz Universität Hannover. Arbeitsgebiete: Computeralgebra, Arrangements von Hyperebenen, Spiegelungsgruppen, Quantengruppen und Tensor kategorien. Entwickler von zahlreichen Computerbeweisen und Experimenten für die Forschung, insbesondere mit Magma, GAP und Sage.

<https://www.iazd.uni-hannover.de/cuntz.html>



**Prof. Dr. Claus Fieker**, Professor für konstruktive Zahlentheorie und Computeralgebra an der TU Kaiserslautern. Arbeitsgebiete: Computeralgebra, konstruktive Zahlentheorie, Klassenkörpertheorie, Darstellungstheorie und Galois Theorie. Mitentwickler von Magma (elf Jahre in Sydney), Kant/Kash und Singular.

<http://www.mathematik.uni-kl.de/~fieker>



**Prof. Dr. Anne Frühbis-Krüger**, ab 1.10.2019 Professorin am Institut für Mathematik der Carl von Ossietzky Universität Oldenburg (vorher Leibniz Universität Hannover), Arbeitsgebiete: Algorithmische Singularitätentheorie, Algorithmische Algebraische und Arithmetische Geometrie, seit 1996 Mitarbeit an der Entwicklung des Computeralgebrasystems Singular.

<http://gandalf.krueger-berg.de/~anne>



**Dr. Thomas Hahn**, wissenschaftlicher Mitarbeiter am Max-Planck-Institut für Physik, München. In der Fachgruppenleitung seit 2002 als Fachexperte Physik, Autor der Computeralgebra-Softwarepakete FeynArts und FormCalc für Rechnungen im Bereich der Teilchenphysik.

<http://www.th.mpp.mpg.de/members/hahn>



**Prof. Dr. Florian Heß**, Professor am Institut für Mathematik der Carl von Ossietzky Universität Oldenburg. Arbeitsgebiete: Algorithmische algebraische Zahlentheorie und Geometrie, speziell algebraische Funktionenkörper, Kurven und Anwendungen auf Kryptographie und Codierungstheorie. Umfangreiche Mitarbeit an den Computeralgebrasystemen Kant/Kash und Magma sowie Tätigkeiten im Bereich der Kryptographie.

<https://uol.de/florian-hess>



**Prof. Dr. Max Horn**, Professor für algorithmische Algebra an der Universität Siegen. Arbeitsgebiete: Computeralgebra (insbesondere Gruppentheorie) algebraische Lie-Theorie, Kac-Moody-Gruppen und Gebäude. Mitentwickler des Computeralgebrasystems GAP.

<https://www.quendi.de/math>



**Prof. Dr. Gregor Kemper**, Professor für algorithmische Algebra an der TU München. Arbeitsgebiete: Invariantentheorie, algorithmische kommutative Algebra, Computeralgebra. Autor von Software-Paketen für Invariantentheorie in Maple und Magma.  
<http://www.groups.ma.tum.de/algebra/kemper>



**Prof. Dr. Jürgen Klüners**, Professor für Computeralgebra und Zahlentheorie an der Universität Paderborn. Arbeitsgebiete: Computeralgebra, Galois- und Zahlentheorie. Mitentwickler der Computeralgebrasysteme Kant/Kash und Magma sowie einer Datenbank für Zahlkörper.  
<https://math.uni-paderborn.de/ag/klueners/>



**Prof. Dr. Martin Kreuzer**, Universitätsprofessor, Lehrstuhl für Symbolic Computation, Fakultät für Informatik und Mathematik, Universität Passau. Arbeitsgebiete: Computeralgebra, insbesondere Gröbnerbasen und Randbasen, industrielle Anwendungen der Computeralgebra, algebraische Kryptographie, algebraische Geometrie. Leiter des Entwicklerteams des Computeralgebrapakets ApCoCoA.  
<http://staff.fim.uni-passau.de/~kreuzer>



**Dr. Fabian Reimers**, wissenschaftlicher Mitarbeiter am Lehrstuhl für algorithmische Algebra, Technische Universität München. Arbeitsgebiet: Invariantentheorie. Seit 2017 als Redakteur des Computeralgebra-Rundbriefs in der Fachgruppenleitung.  
<http://www.groups.ma.tum.de/algebra/reimers>

## Workshop-Förderung der Fachgruppe:

Sie veranstalten einen Workshop zu einem Thema aus dem Bereich der Computeralgebra und könnten mit einer kleinen finanziellen Unterstützung den Workshop deutlich interessanter oder effektiver gestalten? Die Fachgruppe Computeralgebra unterstützt Workshops mit bis zu 1000,- Euro.

Anträge können mit einer kurzen Beschreibung des Workshops (ca. 1 DIN A4 Seite; kurze Beschreibung des Gebiets, Thema des Workshops, Zielgruppe, Budget-Planung) und einer Darstellung, inwiefern diese Förderung einen deutlich erkennbaren Beitrag zum Gelingen des Workshops und zur Nachwuchsförderung liefert, an den Sprecher der Fachgruppe gerichtet werden: **kemper@ma.tum.de**, bitte „**Workshop-Förderung**“ im Betreff angeben.



### Tagung der Fachgruppe Computeralgebra, Kassel, 16.5. – 18.5.2019

<http://www.fachgruppe-computeralgebra.de/kassel-2019>



*Tagung der Fachgruppe 2019 in Kassel*

Von 16. bis 18. Mai 2019 fand die achte Computeralgebra-Tagung der Fachgruppe statt und zwar bereits zum siebten Mal in Kassel unter der bewährten Leitung von Wolfram Koepf. Das Ziel dieser Tagungsreihe ist es, einerseits Nachwuchswissenschaftlern zu ermöglichen ihre Ergebnisse vorzustellen, andererseits aber auch Hauptvortragende zu gewinnen, die Übersichtsvorträge über wichtige Gebiete der Computeralgebra und über Computeralgebra-Software geben. Bei der diesjährigen Ausgabe hielten folgende fünf Wissenschaftler die Hauptvorträge:

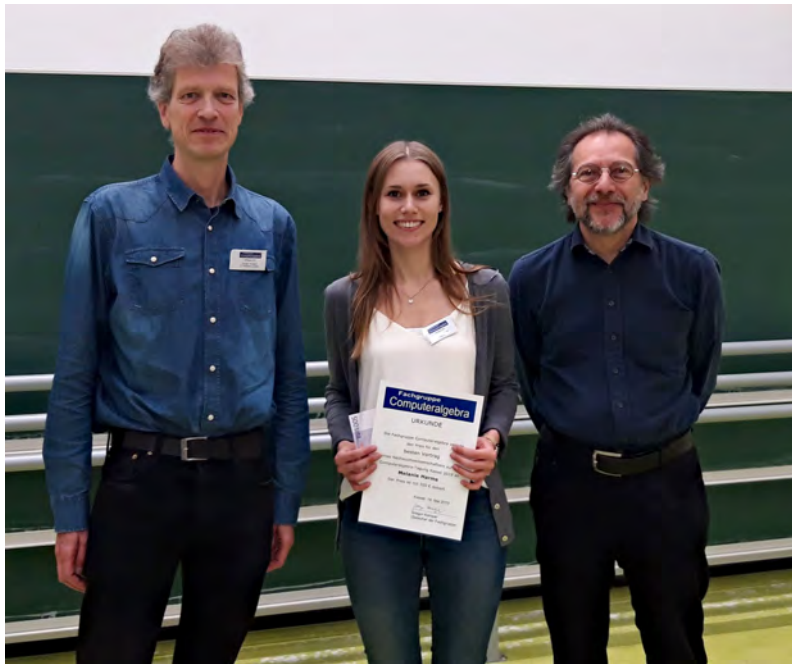
- **Matthias Junge (Oldenburg):** *Asymptotisch schnelle Arithmetik in der Picardgruppe algebraischer Kurven*
- **Markus Kirschmer (Aachen):** *Quaternäre quadratische Formen*
- **Hannah Markwig (Tübingen):** *Ebene tropische Kurven und ihre Berechnung*
- **Bernd Sturmfels (Leipzig):** *Sixty-four Curves of Degree Six*
- **Rebecca Waldecker (Halle):** *Kanonische Bilder.*

Die Tagung wurde am Donnerstagnachmittag mit dem Hauptvortrag von Matthias Junge eröffnet, dem Nachwuchs-Preisträger der Computeralgebra-Tagung 2017. Am Freitag und Samstag gab es je zwei weitere Hauptvorträge. Dazwischen wurden insgesamt 18 halbstündige Vorträge in zwei parallelen Sektionen gehalten. Das Programm wurde durch einen Beitrag von Thomas Richard über neue Features in Maple 2019 ergänzt sowie durch einen Bericht von Gregor Kemper über Neuigkeiten aus der Fachgruppe.

Am Freitagabend wurden die Teilnehmer von einem Bus abgeholt und zur Königs-Alm gefahren. Kaum zu glauben, aber wahr: Hier wurde eine bayrische Almhütte (fast) originalgetreu ins nordhessische Hügelland versetzt. Bei deftigen Schmankerln verging der Abend wie im Fluge. Die Mitglieder der Fachgruppenleitung nutzten die Gelegenheit, um die Nachwuchsbeiträge zu diskutieren und die für den letzten Tag vorgesehene Preisvergabe zu planen. Auch dieses Jahr fiel die Wahl wieder sehr schwer, da die Vorträge durchweg von hohem Niveau waren bei gleichzeitiger Wahrung der Zugänglichkeit für Nichtexperten. Hier hat sich bei der seit 2003 stattfindenden Tagung der Fachgruppe ein Vortragsstil herausgebildet, der einem so heterogenen Gebiet wie der Computeralgebra hervorragend zu Gesichte steht.



Schließlich konnte sich Melanie Harms (Aachen) mit ihrem Vortrag über invariante (semi-)algebraische Mengen polynomieller Differentialgleichungssysteme durchsetzen. Sie erhielt den mit 500 Euro dotierten Preis, der mit der Einladung verbunden ist, bei der nächsten Computeralgebra-Tagung einen Hauptvortrag zu halten.



*Preisvergabe bei der Tagung der Fachgruppe 2019 in Kassel*

Man mag es kaum für möglich halten, aber Wolfram Koepf, der langjährige Organisator der Computeralgebra-Tagung der Fachgruppe, ist 2019 in den sogenannten Ruhestand eingetreten. Es ist daher derzeit völlig offen, wo und unter welcher Leitung die nächste Ausgabe dieser Tagungsreihe stattfinden wird. Darüber wird zu gegebener Zeit zu berichten sein. Momentan gilt es, Wolfram Koepf und seinem Team, insbesondere Frau Martina Syborg, ganz herzlich für ihr großes Engagement zu danken. Weiterhin geht der Dank der Fachgruppenleitung an die Sponsoren und alle weiteren Unterstützer und Helfer.

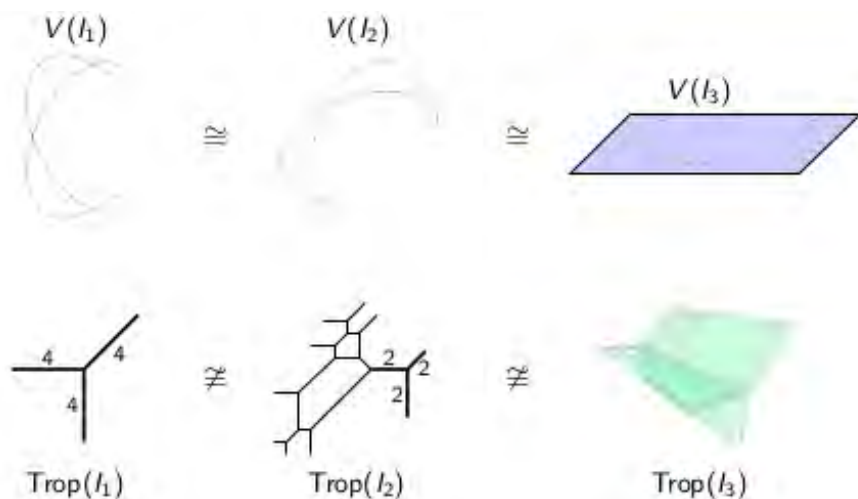
Eva Zerz (Aachen)

# Minisymposium Computeralgebra

DMV Jahrestagung Karlsruhe 23.-26.9.2019

Das Minisymposium 'Computeralgebra' zeigte in seinen sieben Vorträgen sehr schön die Breite der Forschungs- und Einsatzgebiete. Von klassischer algebraischer Geometrie und tropischer Geometrie über arithmetische Geometrie bis hin zu differentieller Galoistheorie und Quantengruppen reichte diesmal das Spektrum der Vorträge. Dabei war der zweite Teil am Dienstagmorgen deutlich besser besucht als der erste am Montagnachmittag, was sicher auch mit dem eher schleppend anlaufenden Besuch der Jahrestagung an sich zusammenhing.

In den Vorträgen wandten sich alle Sprecherinnen und Sprecher ganz bewusst an ein breites Computeralgebra-Fachpublikum und fokussierten sich stärker auf die Vorstellung ihres jeweiligen Forschungsgebietes als auf technische Details. Yue Ren legte in seinem Vortrag über tropische Basen und Tropikalisierungen großen Wert darauf, Unterschiede und Gemeinsamkeiten zwischen tropischer und algebraischer Geometrie hervorzuheben. So sind etwa tropische Varietäten, die von algebraischen Varietäten abstammen, nicht unabhängig von der Wahl des Koordinatensystems und den Erzeugern des ursprünglichen Ideals. Damit stellt sich die Frage nach guten Basen im tropischen Kontext sogar noch dringlicher als im algebraischen. Umgekehrt kann man aber auch fragen, welche tropischen Kurven (mit vorgegebenen Invarianten) sich in eine tropische Ebene einbetten lassen. Mit eingängigen Bildern hinterlegt, erläuterte Ren hierzu den aktuellen Stand der Forschung.



*Ein Schnappschuss aus Yue Rens Vortrag über tropische Basen – oben isomorphe algebraische Varietäten, unten die zugehörigen nicht isomorphen tropischen Analoga.*

Jeroen Sijtsling gab in seinem Vortrag eine Übersicht über Fragen und algorithmische Methoden im Studium von Jacobischen von Kurven, einem Gebiet der arithmetischen Geometrie. Anders als in vielen anderen Zusammenhängen wird hier zur Beschreibung der Objekte nicht auf die zugehörigen Gleichungen zurückgegriffen, die vielfach zu kompliziert sind, um die wesentlichen Aspekte leicht zugänglich zu machen. Stattdessen weicht man hier auf eine komplex-analytische Beschreibung aus, unter Verwendung der sogenannten Periodenmatrix. Mit dieser Beschreibung ist es zum Beispiel möglich, die Endomorphismen der Jacobischen zu bestimmen sowie eine Zerlegung in einfache Faktoren.

Ulrich Thiel führte in das Studium von symplektischen Varietäten ein, die nicht zuletzt durch den engen Bezug zu Phasenräumen in der theoretischen Physik von großem Interesse sind. Genauer ist Thiel an symplektischen Quotientensingularitäten nach endlichen Gruppenoperationen interessiert. Während für Flächen die minimale Auflösung von Singularitäten der Fläche stets symplektisch ist, kann man in höheren Dimensionen nicht mehr von der Existenz einer symplektischen Auflösung ausgehen. Vielmehr ist es hier nur noch sinnvoll nach minimalen Modellen zu fragen, von denen es aber zu einer gegebenen Singularität endlich viele geben kann (gezählt bis auf Isomorphie). Die Erläuterung, wie man deren Zahl bestimmen kann, und warum dies einfacher ist als die Bestimmung aller möglichen minimalen Modelle, schlug dann den Bogen zum aktuellen Stand der Forschung.

Am Dienstagmorgen führte dann Matthias Seiß das Publikum in die differentielle Galoistheorie, die dort auftretenden Fragestellungen sowie deren aktuelle Fortschritte ein. Das grundlegende Setting ist hierbei analog zu dem der klassischen Galoistheorie, wenn auch z.B. mit einem zugrunde liegenden differentiellen Körper und differentiellen Galoisgruppen gearbeitet wird, welche Untergruppen von Matrixgruppen über diesem Körper sind. Differentielle Galoistheorie liefert dann Antworten auf die Frage nach der Lösbarkeit durch Liouville-Erweiterungen.

Durch eine Erläuterung der allgemeinen Methodik für konkrete Gruppen und einen Blick auf dabei verwendete Algorithmen schlug Seiß dann die Brücke zur Computeralgebra.

Jacinta Torres beschäftigte sich in ihrem Vortrag mit der Fragestellung, zu einer gegebenen 'Formel' eine sogenannte Statistik, also eine Abbildung aus einer geeignet gewählten Menge in die ganzen Zahlen, zu finden, die zu der Formel passt. Auch wenn diese Fragestellung erst einmal sehr unkonkret erscheint, tritt sie doch in verschiedenen Zusammenhängen wie der algebraischen Kombinatorik, der Darstellungstheorie und der enumerativen Geometrie immer wieder auf. Konkreter befasste sich Torres im Vortrag mit Statistiken auf Young Tableaux, die es erlauben, geschlossene Formeln für Kostka-Foulkes-Polynome (eine Spezialfall von Kazhdan-Lusztig-Polynomen) zu gewinnen. Während ein Algorithmus zur Behandlung dieser Art von Problemen bereits durch Arbeiten von Lecouvey bekannt war, ermöglicht hier die neue Formulierung als Spiel auf der Struktur der Tableaux einen wesentlich einfacher strukturierten Zugang und auch eine einfachere Implementation.

Moritz Weber versah seinen Vortrag mit dem Untertitel 'durch Computereperimente zu theoretischen Erkenntnissen' und gab eine sehr zugänglich gehaltene Einführung in das Gebiet der Quantengruppen. Konkreter klärte er erst über die Bedeutung des vor einen Begriff gesetzten Wortes 'Quantum' auf, nämlich den Übergang in ein nicht-kommutatives Setting. Insbesondere sind Quantengruppen gar keine Gruppen im klassischen Sinn, sondern wesentlich allgemeinere Objekte. Im Zentrum seines Vortrags standen dann Methoden zur Gewinnung neuer Quantengruppen mittels Computeralgebrawerkzeugen.

Den Abschluss des Minisymposiums bildete der Vortrag von Michael Joswig über die Enumeration von Triangulierungen – was z.B. im Spezialfall der  $n$ -Ecke genau zu den Catalan-Zahlen führt. Das algorithmische Problem bei der Berechnung der Menge aller regulären Triangulierungen endlich vieler Punkte profitiert besonders von massiv parallelem Vorgehen, welches in diesem Kontext in einem Reverse-Search-Algorithmus realisiert ist. Eine zusätzliche Analyse des Settings zeigt zudem, dass man sich in einer atypischen Ausgangslage für den Schreier-Sims-Algorithmus befindet, in welcher eine neue Variante zu einer deutlich schnelleren Berechnung von Orbit-Repräsentanten führt.

Gerade der Anteil junger Vortragender, das engagierte Eingehen der Vorträge auf ein recht diverses Publikum und auch einige neu vertretene Gebiete machten den besonderen Reiz dieses Minisymposiums aus. Die räumlichen Gegebenheiten und die technische und organisatorische Unterstützung durch die Organisatoren der Jahrestagung kann als optimal gelten. Hierfür geht unser Dank an die Kollegen vom KIT.

Im selben Jahr ein Minisymposium auszurichten, in dem auch die Computeralgebra-Tagung der Fachgruppe in Kassel stattfand, war vielleicht ein Wagnis, das sich aber in den Augen der Teilnehmer und Ausrichter mehr als gelohnt hat.

Anne Frühbis-Krüger (Oldenburg)

### Isogeny-based cryptography

Chloe Martindale, Technische Universiteit Eindhoven  
Lorenz Panny, Technische Universiteit Eindhoven

chloemartindale@gmail.com  
lorenz@yx7.cc



---

### Introduction

---

*Quantum computers* threaten to break most of the cryptography we are currently using to secure critical computer systems such as the internet. A quantum computer is a machine which employs quantum-physical phenomena to perform computations in a way that's fundamentally different from a “normal”, *classical*, computer. Whereas a classical computer is, at any point in time, in a fixed *state* — such as a bit string representing its memory contents — the state of a quantum computer can be a “mixture”, a so-called *superposition*, of several states. Note that the internal state is *hidden*: The only way to get information about the state is to perform a *measurement*, which will return a single non-superimposed *classical* output, such as a bit string, that is *randomly distributed according to the internal state*, and the internal state gets replaced by the measurement outcome. For example, when measuring an equal superposition of the two-qubit states  $|00\rangle$  and  $|11\rangle$ , the result would be one of the bit strings 00 or 11 with probability  $1/2$  each. Now a *quantum algorithm* consists of applying a carefully crafted sequence of operations to the internal state of a quantum computer in order to amplify the desired piece of information in the superposition, followed by measurements to extract the result.

The extra computational power thanks to the ability to store and manipulate *superpositions* of states allows for more efficient algorithms to tackle some computational problems. Note that contrary to a common misconception, quantum computers are *not* known to provide massive speedups over classical computers for “many”, or even “all”, tasks; in fact, there is only a handful of problems where known quantum algorithms outperform the best currently known classical algorithms. Unfortunately, many of these problems are at the heart of today's cryptographic systems; we shall see an example of this in the next section.

To deal with this problem, researchers have come up with *post-quantum cryptography*, a set of proposals for solutions to the looming threat of quantum computers on the cryptography currently in use. It may seem

that quantum computers effective enough to break real-world encryption are a long way off: Building a quantum computer with enough qubits, and keeping them stable for long enough to be useful, poses a set of incredibly difficult physics and engineering challenges. However, no matter whether one believes powerful quantum computers are five years off or thirty years off, there is a compelling argument for acting as early as possible: People are now sharing plenty of data via cryptographically secured channels that they intend to stay private forever — or at least for a very long time —, even when stored now and attacked later with a quantum computer. This includes online audio or video telephony, private messages sent through chat services such as WhatsApp, and other sensitive data such as medical or financial records. In the words of a recent report by the United States' National Academy of Sciences:

Even if a quantum computer that can decrypt current cryptographic ciphers is more than a decade off, the hazard of such a machine is high enough — and the time frame for transitioning to a new security protocol is sufficiently long and uncertain — that prioritization of the development, standardization, and deployment of post-quantum cryptography is critical for minimizing the chance of a potential security and privacy disaster. [13]

*Isogeny-based cryptography* is a specific type of post-quantum cryptography that uses certain well-behaved maps between abelian varieties over finite fields (typically elliptic curves) as its core building block. Its main advantages are relatively small keys and its rich mathematical structure, which poses some extremely interesting questions to cryptographers and computer algebraists.

The Autumn 2018 issue of this *Rundbrief* contained an article on post-quantum cryptography giving an overview of the main families of proposed constructions. We refer interested readers there for a broader discussion of things happening in the field of post-quantum cryptography.

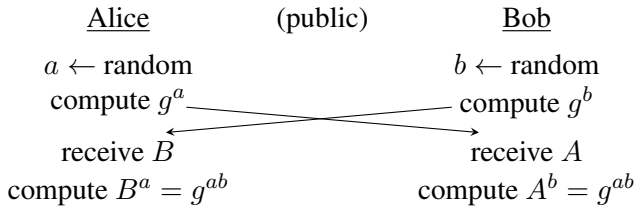


## Classical cryptographic key exchange

An important building block of many cryptographic systems, including the ubiquitous TLS protocol used to secure communication with websites, is a secure *key exchange*. Imagine you and a friend want to be able to send each other messages in a (metaphorical) locked box over an insecure channel, so you both need the same key. Since it's often not practical to exchange keys in person with everyone you want to send messages to, you also need a way of agreeing on a shared secret key over an insecure channel, with nobody else besides you and your friend being able to figure out what that key is. The key can be anything, so long as it's the same for all the parties involved; typically it is encoded as a bit string.

There are several ways to do this, the most common being the *Diffie–Hellman key exchange*. The security of this method is based on the fact that, for a finite group  $G$ , if you share an element  $g \in G$  and some randomly chosen power  $g^a \in G$ , it is in general very hard for a corrupt bystander to compute  $a$ . The traditional (but now mostly deprecated) instantiation consists of fixing a prime  $p$  and computing in the group  $\mathbb{F}_p^*$ , i.e., the element  $g$  is simply an integer and the power  $g^a$  is computed modulo  $p$ .

To use this operation for a key exchange, our two parties Alice and Bob first agree on a group  $G$  and an element  $g \in G$ . Alice then chooses a secret positive integer  $a$ , and Bob chooses a secret positive integer  $b$ . Together, they can then compute the value  $g^{ab}$  over a public channel as follows:



**Figure 1:** The Diffie–Hellman key exchange.

The obvious way to attack this scheme is to recover one of the secrets  $a$  and  $b$  from the publicly transmitted values  $g^a$  and  $g^b$ . This is known as the *discrete-logarithm problem*, which appears to be computationally hard for classical computers when the group  $G$  is well-chosen.

However, unfortunately, one thing that quantum computers are particularly good at is finding *periods* of computable functions using (variants of) an algorithm by Shor [11], which can be used to attack the discrete-logarithm problem as follows. Note that public values  $g^x$  are simply group elements: They can be multiplied together, and this satisfies the rule  $g^x \cdot g^y = g^{x+y}$ . This is exactly the operation that reduces breaking the scheme to finding a period: Given the element  $g$  and a public key  $A = g^a$ , we can define the group homomorphism

$$f: \mathbb{Z}^2 \rightarrow G, (x, y) \mapsto g^x \cdot A^y = g^{x+ay}.$$

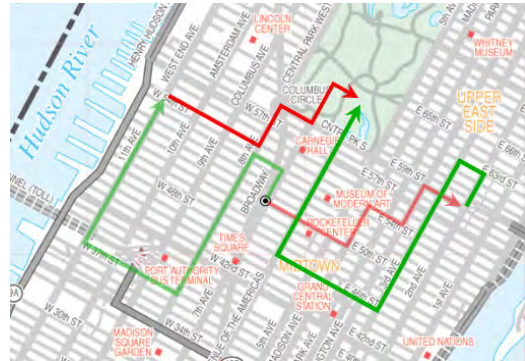
Hence  $f$  is a periodic map whose period lattice is just its *kernel*, i.e., those pairs  $(x, y) \in \mathbb{Z}^2$  where  $g^{x+ay} = 1$ .

Shor's algorithm can, in polynomial time, find a basis of this lattice from an efficient description of the map  $f$ . We can then look for a vector of the form  $(\tau, -1)$  in the lattice, which must equal  $(a, -1)$  modulo the order of  $g$ , thus we have found  $a$ . The bottom line is that Diffie–Hellman is broken by quantum computers in all groups. Is this the end?

Luckily, in the aftermath of the discovery of Shor's algorithm, the traditional Diffie–Hellman framework has been extended to schemes which have similar traits, but do not rely on exponentiation maps in groups being one-way. One of these variants uses *isogeny graphs*.

## Key exchange from graphs

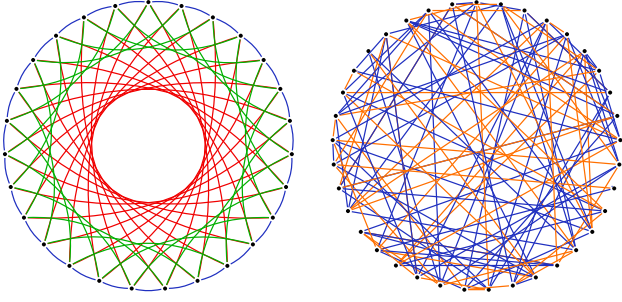
Before we talk about isogeny graphs, let's first see how to get a shared value (key) from a more familiar graph. Here “graph” refers to a collection of *nodes* (dots) and *edges* (lines between them). For example, we can create a graph from a map of Manhattan by drawing a node at every junction and drawing an edge for every street. Then if Green and Red want to compute a shared value, they each choose a (secret) path from a common starting point, share the coordinates of their endpoints, then follow the same path from each other's endpoints to end up at the same final coordinates.



**Figure 2:** Diffie–Hellman in Manhattan.

However, this clearly isn't a secure way of exchanging keys—anyone can find a path from the common starting point to either Green or Red's end-of-path coordinates and thus compute the shared final coordinates, literally by adding and subtracting. To turn this approach into a secure key exchange, we need to replace our graph of Manhattan with a less structured graph, one in which finding a path between two given nodes is infeasible. On the other hand, the graph still needs to have *enough* structure to allow composing paths in a meaningful, commutative way, such that both parties end up at the same spot.

This is where isogenies comes in: They give rise to two families of graphs which are believed to have all the required properties for a (post-quantum) key exchange. Typical examples of these graphs look like this:

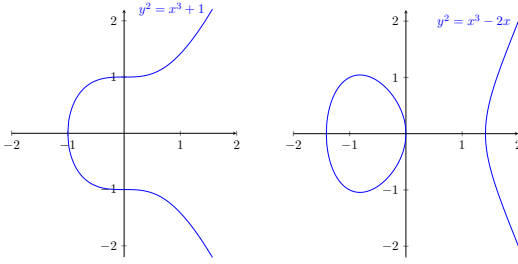


**Figure 3:** Special isogeny graphs over a finite field.

In both graphs, each node represents an *elliptic curve*, which can be represented as a certain kind of polynomial, and the edges represent maps between the elliptic curves called *isogenies*.

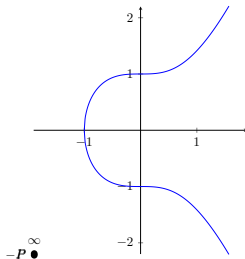
## Elliptic curves and isogenies

We shall now explain some of the necessary background for understanding isogeny-based cryptography. Let  $p \geq 5$  be a prime. An *elliptic curve over  $\mathbb{F}_{p^k}$*  can then be defined as a smooth curve with equation  $E: y^2 = f(x)$ , where  $f(x)$  is a degree-3 polynomial with coefficients in  $\mathbb{F}_{p^k}$ .



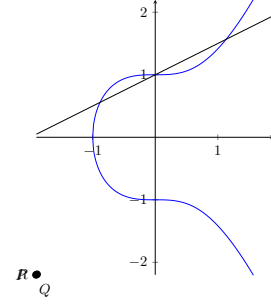
**Figure 4:** Two elliptic curves over  $\mathbb{R}$ .

“Smooth” means that the graph does not intersect itself or have any sharp points (“cusps”). These equations are especially interesting because the solutions that are defined over  $\mathbb{F}_{p^k}$ , together with one extra element, form an abelian group denoted  $E(\mathbb{F}_{p^k})$ . The extra element is referred to as the *point at infinity*  $\infty$  and is defined to be lying on every vertical line that intersects the curve. The point  $\infty$  is also the identity element of the group. The inverse of a point (a solution  $P = (x_0, y_0)$  to the defining polynomial of  $E$ ) is defined to be  $-P = (x_0, -y_0)$ . Notice that on a vertical line that intersects the elliptic curve in a point  $P = (x_0, y_0)$ , there are a total of three points in the group of solutions to the polynomial of  $E$ : the point  $P$ , its inverse  $-P$ , and the point at infinity  $\infty$ .



**Figure 5:** Negating a point.

In fact, this is no coincidence: any straight line that intersects the elliptic curve will intersect it exactly three times (when counting tangents as intersecting twice and also taking  $\infty$  into account). We use this fact to define the group operation on  $E(\mathbb{F}_{p^k})$ : Given any two solutions  $P = (x_P, y_P)$  and  $Q = (x_Q, y_Q)$ , draw the straight line passing through  $P$  and  $Q$ . This line will intersect the elliptic curve in exactly one more point  $R$ , and it turns out that the coordinates of this point will also be defined over  $\mathbb{F}_{p^k}$ . We then define a *group law*, written as  $+$ , by requiring that  $P + Q + R = \infty$ , the neutral element.



**Figure 6:** Adding points.

This group structure has led number theorists and geometers to study interesting properties of elliptic curves for centuries, and more recently elliptic curves have also enticed cryptographers, most importantly because one can base a very compact and efficient Diffie–Hellman key exchange on it. We need a little more though for a post-quantum scheme, since Shor’s quantum algorithm to compute discrete logarithms of course also applies to this group.

Especially important in isogeny-based cryptography is a specific subclass of elliptic curves: *Supersingular elliptic curves*. An elliptic curve  $E$  defined over  $\mathbb{F}_{p^k}$  is supersingular if  $p \mid (p^k + 1 - \#E(\mathbb{F}_{p^k}))$ . The most important special cases that come up are  $E$  defined over  $\mathbb{F}_p$  with  $\#E(\mathbb{F}_p) = p + 1$ , and  $E$  defined over  $\mathbb{F}_{p^2}$  with  $\#E(\mathbb{F}_{p^2}) = (p + 1)^2$ . In a nutshell, this is useful because it allows to easily enforce a special group order: Given any prime  $p$  and  $k \in \{1, 2\}$ , it is known how to generate a supersingular elliptic curve with  $(p + 1)^k$  points over  $\mathbb{F}_{p^k}$ , hence we can control the group structure by choosing  $p$  in a special way. By contrast, it is not generally known how to efficiently find an elliptic curve with a given number of points, except in particularly nice special cases.

Recall that each node in the graphs we want to use for our post-quantum key exchange represents an elliptic curve. Since these graphs also have edges we need a way of passing from one node to another, which naturally will be a rational map that maps one curve to the other, and we will also want these maps to preserve the group structure of the elliptic curves. An *isogeny* is a non-zero map between elliptic curves that satisfies these things. More precisely, it is a surjective morphism of abelian varieties with finite kernel. The kernel subgroup is of utmost importance: In fact, one can prove that a (separable) isogeny is essentially uniquely defined by its

kernel subgroup, and one can compute an isogeny from its kernel in time linear in the size. Every isogeny has a *degree*, and typically (for separable isogenies) the degree is equal to the size of the kernel. Thus in a sense, the degree quantifies the algebraic and algorithmic complexity of an isogeny. However, since the secret keys in our cryptosystems are isogenies, we will use isogenies with “crypto-sized” (big) degrees! So how do we compute these isogenies quickly?

The solution is to use an isogeny of very *smooth* degree, say  $\deg \varphi = \ell^k$  for a small prime  $\ell$  (less than a few hundred or so) that is coprime to the (usually big) characteristic, and factor it into a composition of much smaller prime-degree maps:

$$E \xrightarrow{\psi_1} E_1 \cdots \rightarrow E_{k-1} \xrightarrow{\psi_k} E'$$

$\varphi$

**Figure 7:** Decomposing a smooth-degree isogeny.

Note that the sequence  $(\psi_1, \dots, \psi_k)$  can be computed in  $O(k \cdot \ell^2)$  field operations, whereas naïvely computing the entire map  $\varphi$  all at once would take time  $\Theta(\deg \varphi) = \Theta(\ell^k)$ : *exponentially* more.

The mathematics behind all of this is much richer than we can show in this short article. Interested readers are kindly referred to Luca De Feo’s lecture notes [4].

## Key exchange on isogeny graphs

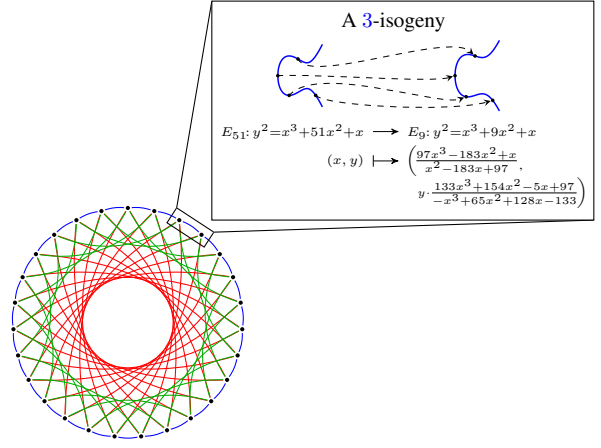
In Figure 3 we saw two very different isogeny graphs that are used in cryptographic protocols. The two protocols built on these two types of graphs are called *CSIDH* [2] (pronounced “seaside”, stands for “Commutative Supersingular-Isogeny Diffie–Hellman”) and *SIDH* [8, 7] (pronounced as individual letters, stands for “Supersingular-Isogeny Diffie–Hellman”).

### CSIDH

We will show the CSIDH key exchange on a small (definitely not cryptographically-sized) example. We have seen how to perform an (insecure) key exchange on the graph on Manhattan — CSIDH is an implementation of the same idea on an *isogeny graph* with:

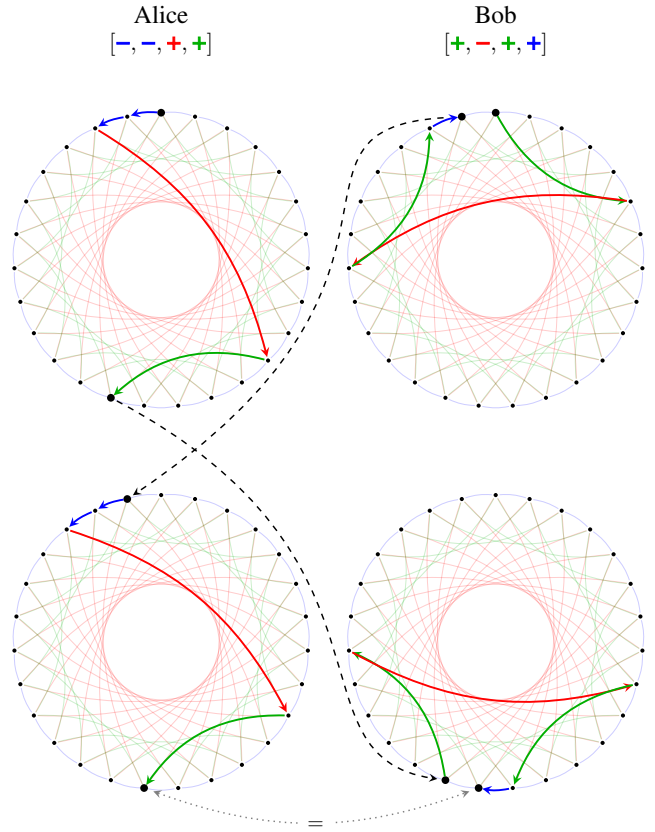
- Nodes given by *supersingular elliptic curves*  $E_A$  with equation  $y^2 = x^3 + Ax^2 + x$  for  $A \in \mathbb{F}_{419}$ .
- Edges given by 3-, 5-, and 7-isogenies.

Almost as though by magic, this graph turns out to be very structured: Every node has exactly two outgoing edges of each colour, and the resulting cycles are compatible in the sense that a red step is always equivalent to the same number of blue steps, etc., independent of the position:



**Figure 8:** An isogeny graph with a zoomed-in edge.

Now if Alice and Bob want to compute a shared value in this graph, they each choose a (secret) path from a common starting point, share their endpoints, then follow the same path from the respective other party’s endpoint to end up at the same final node — just like in the Manhattan example. We describe a path by a list of directions: one step clockwise (+) or anticlockwise (−) on the sub-graph of a given colour.



**Figure 9:** Key exchange on a regular isogeny graph.

On an unstructured graph, there is a priori no reason why Alice and Bob would end up at the same node after following this “graph key exchange” method. So what is it about the structure of *this* graph that makes this work?

Consider the set of clockwise and anticlockwise  $\ell$ -isogenies as  $\ell$  ranges over the non-negative integers, taken up to isomorphism. This set forms a *commuta-*



tive group  $G$  that acts on the set of supersingular elliptic curves  $E_A: y^2 = x^3 + Ax^2 + x$  with  $A \in \mathbb{F}_p$ .<sup>1</sup> Write  $f_\ell^+$  and  $f_\ell^-$  for clockwise and anticlockwise  $\ell$ -isogenies respectively. In our key-exchange example above, Alice computes her curve  $E_A$  by computing the action of

$$f_A = f_7^+ \cdot f_5^+ \cdot f_3^- \cdot f_3^-$$

on  $E_0$ , written as  $E_A := f_A * E_0$ , and Bob computes his elliptic curve  $E_B$  by computing the action of

$$f_B = f_7^+ \cdot f_5^- \cdot f_7^+ \cdot f_3^+$$

on  $E_0$ , written as  $E_B := f_B * E_0$ . Then Alice and Bob send each other  $E_A$  and  $E_B$  and compute the actions  $f_A * E_B$  and  $f_B * E_A$  respectively. Now, as  $f_A$  and  $f_B$  are both elements in a commutative group,

$$\begin{aligned} f_A * E_B &= (f_A \cdot f_B) * E_0 \\ &= (f_B \cdot f_A) * E_0 = f_B * E_A. \end{aligned}$$

So Alice's endpoint  $f_A * E_B$  and Bob's endpoint  $f_B * E_A$  are the same!

Observe that, during this exchange, Alice never needs to communicate the isogeny group element  $f_A$ ; this is her *private key*. With a much larger example, i.e., replacing 419 by a prime of several hundred (or even thousand) bits, and using a much longer list of  $\ell$ s, recovering this secret isogeny group element given just the start and end curve becomes infeasible for an attacker.

You may be thinking: doesn't Shor's algorithm apply to groups? Can I attack this commutative "isogeny group" with a quantum computer? The fundamental difference is that in traditional Diffie–Hellman, the public keys themselves are elements of the group, whereas in CSIDH, only the private keys are elements of a group — the public keys are elements of a set on which the group acts, and the operation  $g^x \cdot g^y = g^{x+y}$  we've seen exploited earlier to apply Shor's algorithm to the discrete-logarithm problem simply does not exist. However, there is a quantum algorithm due to Kuperberg which attacks the action of a commutative group on the set of public keys to get a *subexponential* (but still super-polynomial) quantum attack [9, 10, 3]. In practice, this means that in order to be post-quantum secure, the parameters (like the prime  $p$ ) have to be chosen larger than if the best attack was exponential; exactly how much larger is an ongoing research question.

## SIDH

On the second graph in Figure 3, there is no evident group action — it looks just random. This rightfully suggests that Kuperberg's algorithm may not apply to finding a path on this graph, so the security level might scale better. However, it is not obvious how to even

make a key-exchange protocol *work* on this graph: The extremely regular structure of the CSIDH graph aided in getting Alice and Bob's operations to commute, whereas in this case everything looks rather messy.

Happily, the graph does still carry enough structure, due to the fact that an isogeny is uniquely defined by its kernel. Alice and Bob (publicly) agree on a common starting curve  $E/\mathbb{F}_{p^2}$  and choose secret subgroups  $A$  and  $B$  of  $E(\mathbb{F}_{p^2})$ . Writing  $\varphi_A$  and  $\varphi_B$  for the isogenies with those subgroups as kernel, the following diagram commutes (up to isomorphism) when  $A' = \varphi_B(A)$  and  $B' = \varphi_A(B)$ , and therefore an isomorphism invariant of the curve  $E/\langle A, B \rangle$  can be used as a shared secret:

$$\begin{array}{ccc} E & \xrightarrow{\varphi_A} & E/A \\ \downarrow \varphi_B & & \downarrow \varphi_{B'} \\ E/B & \xrightarrow{\varphi_{A'}} & E/\langle A, B \rangle \end{array}$$

Figure 10: High-level view of SIDH.

Here, the horizontal arrows are computed by Alice and the vertical arrows are computed by Bob. Focussing on Alice, the only problem left is that she needs to somehow obtain  $A' = \varphi_B(A)$ , but Bob cannot give out  $\varphi_B$  since that's his secret.

The solution is that isogenies are also group homomorphisms on the corresponding groups of elliptic curve points: While Bob cannot give out  $\varphi_B$ , he *can* evaluate this map on publicly known points  $P, Q \in E(\mathbb{F}_{p^2})$  and reveal  $P' = \varphi_B(P)$  and  $Q' = \varphi_B(Q)$ . If Alice then chooses her subgroup  $A$  of the form  $\langle P + [a]Q \rangle$  (that is, a cyclic group generated by  $P + [a]Q$ ), she can simply compute  $A'$  as  $\langle P' + [a]Q' \rangle$ .

Similarly, Alice publishes images of known points under her own secret  $\varphi_A$ , allowing Bob to find  $B'$ .

This is the high-level mathematical overview of the protocol — of course there are many more interesting details in practice, for example one still has to ensure that the points  $P', Q'$  do not leak computationally useful<sup>2</sup> information about  $\varphi_B$  (similarly for  $\varphi_A$ ), and choose the other parameters of the system in such a way that everything Alice and Bob need to compute is efficient in practice.

## More advanced protocols

In this article we've only shown two simple key-exchange protocols using different kinds of isogeny graphs. However, the underlying mathematical ideas give rise to many other interesting cryptographic constructions, some of which seem impossible or harder to build without the use of isogenies.

<sup>1</sup>For number theory experts: a (separable) isogeny is (up to isomorphism) uniquely defined by its kernel, which corresponds to an ideal in the common  $\mathbb{F}_p$ -rational endomorphism ring  $\mathbb{Z}[\sqrt{-p}]$  of every such elliptic curve. The *codomains* of such isogenies then only depend on the *class* of the corresponding ideal, hence the action of  $G$  can be considered an action of (a subgroup of) the class group  $\text{cl}(\mathbb{Z}[\sqrt{-p}])$ .

<sup>2</sup>We emphasize that while the action on a set of points often uniquely identifies an isogeny, it is not generally known how to *compute* the isogeny from that information.



For instance, one can build a *verifiable delay function* from isogenies [6]: A VDF is a random-looking function which is inherently slow to compute — independently of the algorithms used, the amount of hardware available, and parallelism — but on the other hand it is efficient for anyone to verify afterwards that the result is correct. (In the isogeny-based construction, the slow part consists of a long isogeny evaluation, and the verification part is a single elliptic-curve pairing.) This functionality may seem like not more than an odd curiosity, but in fact it has enough relevance in the distributed-systems world that blockchain projects are currently investing one hundred thousand US dollars [14] in the development of VDF technology (albeit founded on different mathematical ideas).

Another interesting example is the construction of *digital signatures* from isogenies: A recent paper [1] proposes a practical signature scheme CSI-FiSh (pronounced “seafish”, stands for “Commutative Supersingular-Isogeny Fiat–Shamir”) based on CSIDH’s 512-bit parameter set. The main contribution is a massive precomputation effort in the form of a record-breaking *class-group computation*, which allows uniform sampling of isogeny walks — an important ingredient of the signature scheme. It is not known how to adapt this scheme to bigger (read: more secure) parameters: the effort for the class-group computation quickly grows too big for currently known techniques.

## References

- [1] Ward Beullens, Thorsten Kleinjung, and Frederik Vercauteren. CSI-FiSh: Efficient isogeny based signatures through class group computations. In *Cryptology ePrint Archive*, Report 2019/498. 2019. <https://ia.cr/2019/498>.
- [2] Wouter Castryck, Tanja Lange, Chloe Martindale, Lorenz Panny, and Joost Renes. CSIDH: an efficient post-quantum commutative group action. In *ASIACRYPT (3)*, volume 11274 of *LNCS*, pages 395–427. Springer, 2018. <https://ia.cr/2018/383>.
- [3] Andrew M. Childs, David Jao, and Vladimir Soukharev. Constructing elliptic curve isogenies in quantum subexponential time. In *J. Mathematical Cryptology*, volume 8, pages 1–29. 2014. <https://arxiv.org/abs/1012.4019v1>.
- [4] Luca De Feo. Mathematics of isogeny based cryptography. 2017. <https://arxiv.org/abs/1711.04062>.
- [5] Luca De Feo and Steven D. Galbraith. SeaSign: Compact isogeny signatures from class group actions. In *EUROCRYPT (3)*, volume 11478 of *LNCS*, pages 759–789. Springer, 2019. <https://ia.cr/2018/824>.
- [6] Luca De Feo, Simon Masson, Christophe Petit, and Antonio Sanso. Verifiable Delay Functions from supersingular isogenies and pairings. In *Cryptology ePrint Archive*, Report 2019/166. 2019. <https://ia.cr/2019/166>.
- [7] David Jao, Reza Azarderakhsh, Matthew Campagna, Craig Costello, Luca De Feo, Basil Hess, Amir Jalali, Brian Koziel, Brian LaMacchia, Patrick Longa, Michael Naehrig, Joost Renes, Vladimir Soukharev, and David Urbanik. SIKE. Submission to [12]. <http://sike.org>.
- [8] David Jao and Luca De Feo. Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. In *PQCrypto*, volume 7071 of *LNCS*, pages 19–34. Springer, 2011. <https://ia.cr/2011/506>.
- [9] Greg Kuperberg. A subexponential-time quantum algorithm for the dihedral hidden subgroup problem. *SIAM J. Comput.*, 35(1):170–188, 2005. <https://arxiv.org/abs/quant-ph/0302112>.
- [10] Greg Kuperberg. Another subexponential-time quantum algorithm for the dihedral hidden subgroup problem. In *TQC*, volume 22 of *LIPIcs*, pages 20–34. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2013. <https://arxiv.org/abs/1112.3333>.
- [11] Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. In *SIAM Journal on Computing*, volume 26(5), pages 1484–1509. 1997.
- [12] National Institute of Standards and Technology. Post-quantum cryptography standardization, December 2016. <https://csrc.nist.gov/Projects/Post-Quantum-Cryptography/Post-Quantum-Cryptography-Standardization>.
- [13] National Academy of Sciences, Engineering, and Medicine. Quantum computing: Progress and prospects. ISBN: 978-0-309-47969-1. 2019. <https://nap.edu/catalog/25196>.
- [14] VDF Alliance. FPGA design competition. See <https://vdfalliance.org/contest>.

# Towards Post-Quantum Secure Symmetric Cryptography: A Mathematical Perspective

Xenia Bogomolec, Quant-X Security & Coding, Hanover  
John Gregory Underhill, itk AVtobvS SARL, Fribourg  
Stiepan Aurélien Kovac, QRCrypto SA, Fribourg

xb@quant-x-sec.com  
john.underhill@protonmail.com  
contact@qrcrypto.ch

## Introduction

We introduce an independent research project on symmetric cryptography with a focus on foreseeable industrial needs. It was initiated by the independent IT-Security experts Kovac and Underhill. The result is the new symmetric cryptographic algorithm EAES, which is intended to be a stronger brother of the widely used AES algorithm (Advanced Encryption Standard), the standardized version of the RIJNDAEL algorithm.

The algorithm EAES is designed by Kovac and Underhill. They published the e-print [1]. Underhill had previously started the implementation in his CEX-NET project, within which he created the cryptographic C++ library CEX [2]. The library is open source and published under the GPL-license. Bogomolec is responsible for the mathematical analysis of the algorithm. All of us are independent IT-Security experts.

EAES mitigates threats by formerly published attacks on AES from binary devices [3, 4, 5] and additionally offers enhanced security against attacks performed by moderate quantum computers [6, 7].

We also outline the necessary considerations and the steps which have to be taken in order to place a new cryptographic algorithm in global industry.

## The importance of standardization

A successful standardization of an algorithm ensures compatibility with other global standards. Furthermore it is a seal of quality for users who don't understand the mechanisms and properties of the algorithm in depth.

There are two big players in international standardization for symmetric cryptography, ISO (International Organization for Standardization) and NIST (National Institute of Standards and Technologies, USA). The NIST launched a standardization process on asymmetric post-quantum cryptography, i.e. cryptography resisting

quantum computing attacks. The reason for the anticipation of standardization to the availability of strong quantum computing machines is very well explained in the article *Isogeny-based cryptography*.

Hybrid encryption systems are used in all major crypto protocols: TLS, SSH and PGP. They combine the advantages of both cryptography classes. Symmetric protocols allow securely sharing a key via digital connections, and symmetric protocols are about  $10^5 \times$  faster than asymmetric ones. The secret session key  $SK$ , which is limited in time, is shared via asymmetric cryptography, and the message itself is symmetrically encrypted with the securely shared session key.

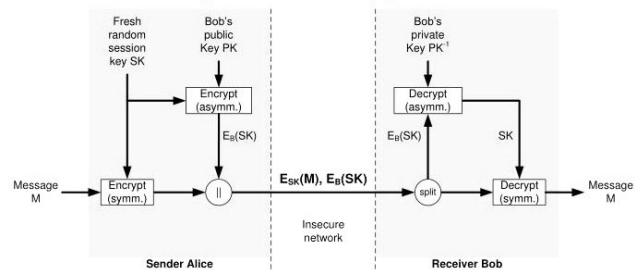


Figure 1: Hybrid encryption.

No asymmetric encryption within hybrid encryption systems can outbalance weaknesses of the symmetric part. ISO currently makes it possible to standardize new symmetric cryptography algorithms and to amend existing ones. Therefore, we strive to establish EAES as an ISO-Standard. Its predecessor AES has been standardized by both organizations.

## Published attacks on AES

There are various types of attacks on cryptographic algorithms, amongst which side-channel attacks, implementation attacks, brute-force attacks and cryptanalysis attacks on AES are of importance in our context. In some cases, those attack types can be combined to

achieve a more efficient decryption of a ciphertext.

**Side-channel attacks** use unintended side effects of cryptographic operations to glean information about the plaintext and/or secret key being processed. **Implementation attacks** use weaknesses in implementation of an encryption scheme, e.g. weak key generation. **Brute-force attacks** attempt every possible combination for a key. **Cryptanalysis attacks** rely on alternative algorithms for finding a secret key of an encrypted text. The latter can be applied to stationary data. Therefore it is interesting for data collectors who are patient enough to wait for the availability of potent-enough machines to perform such an attack.

### Mathematical structure of AES

All RIJNDAEL functions are linear. Only the basic encryption function *SubBytes* (see section Basic RIJNDAEL encryption functions) is often referred to as the non-linear part of AES, but in fact it is linear as well. Well-chosen linear layers with very strong diffusion properties protect against conventional attacks using statistical properties of a cryptosystem. In the case of the quantum algebraic attack [7], this effect is limited.

### Attacks performed on binary devices

Various published attacks on AES [3, 4, 5] take advantage of the invertibility of the original RIJNDAEL key schedule besides other properties such as the relatively low number of rounds, implementation weaknesses, and the same block size for standardized all key sizes.

### Grover's search

AES was generally still considered post-quantum secure for key sizes larger than 192. Even Grover's search algorithm [8] is not regarded a threat for AES-256 in the near future. It can be used to extract the key from a small number of AES plaintext-ciphertext pairs (5 for AES-256). Grover's search is a alternative algorithm for an exhaustive key search (brute-force attack).

### Quantum algebraic attack

A harder impact on the security of AES-256 is posed by the quantum algebraic attack on cryptosystems which can be reduced to Boolean equation solving [7]. This attack reduces security level of AES-256 from 256 to 78.53. The quantum algebraic attack is a classical cryptanalysis attack.

### Consequences

Both Grover's search algorithm and the quantum algebraic attack can be applied to collected data without the corresponding key exchanges as soon as potent-enough quantum computers are available.

Therefore we propose EAES as a symmetric alternative for AES, with higher security against all previously mentioned attacks [3, 4, 5, 6, 7].

The following sections are written for readers with deeper technical knowledge in cryptography and block

ciphers. In the last section we summarize the advantages of EAES.

---

## RIJNDAEL, the base of EAES

---

Here we only give a high-level overview of RIJNDAEL in order to classify our modifications. For a detailed description of its standardized version AES, we refer to the Federal Information Processing Standards Publication 197 [9].

### Computation grounds

RIJNDAEL is an iterative rounds-based block cipher. It relies on a substitution-permutation network. In AES, the network is operating on a  $4 \times 4$  column-major order array of the 128 bits (block size), called the "state." Each input data block is initially transformed into a "state." The term "state" refers to the digital representation as well as to the fact of the continuous transformation during the encryption. So each array entry consists of 8 bits. Columns of the matrix are also called "words." All operations are performed in the Galois field  $GF(2^8) \cong \mathbb{F}_2[x]/(x^8 + x^4 + x^3 + x + 1)$ .

### RIJNDAEL algorithm architecture

RIJNDAEL is a block cipher. That means that the basic encryption functions are iterated a certain number of rounds with dedicated round keys over the state. With  $r$  = number of rounds, the encryption process is:

#### (1) Derive round keys

Call *ExpandKey*(key) to derive  $r + 1$  round keys from the secret key. The  $+ 1$  stands for the fact that *AddRoundKey* is called additionally before each block encryption.

#### (2) Encryption

Perform on each block of data represented by the "state":

##### a) Initial round:

Add the plaintext to the state  
*AddRoundKey*(state, 1st round key)

##### b) For ( $i = 2; i \leq r - 1, i++$ ):

*SubBytes*(state)  
*ShiftRows*(state)  
*MixColumns*(state)  
*AddRoundKey*(state,  $i$ -th round key)

(4 basic RIJNDAEL encryption functions)

##### c) The final round:

*SubBytes*(state)  
*ShiftRows*(state)  
*AddRoundKey*(state, last round key)

The decryption process is composed by the inversed chain of the encryption functions. A symmetric encryption algorithm needs to be composed by invertible functions, but this property wouldn't be necessary for the key expansion. In fact, the invertibility of *ExpandKey* opened the door for various side-channel attacks.

## Basic RIJNDAEL encryption functions

The 4 basic functions of the RIJNDAEL encryption are:

- 1) *AddRoundKey* – addition in  $(\mathbb{F}_2)^{128}$ :  
Bitwise addition of the state and the correspondent round key.
- 2) *SubBytes* – non-linear substitution:  
Each byte is replaced by another according to the specified substitution table (*S*-Box). A more resource friendly option is to treat a state byte as an element  $\alpha \in \mathbb{F}_2[x]/(x^8 + x^4 + x^3 + x + 1)$ , where the multiplicative inverse of  $\alpha$  needs to be found.
- 3) *ShiftRows* – transposition for diffusion:  
The second, third and fourth row of the state are shifted to the left, by 1, 2 and 3 steps.
- 4) *MixColumns* – mixing for diffusion:  
Multiplication of each column of the state with the following matrix  $M$ :

$$M = \begin{bmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{bmatrix}$$

As a chain of invertible algebraic functions, RIJNDAEL and AES can be represented as a polynomial system [10]. This polynomial system can be reduced to a Boolean equation system. We will look at this property in the context of the quantum algebraic attack [7].

## Modifications for EAES

Our modifications affect the key schedule *ExpandKey* and the number of rounds per version. Furthermore we implemented a version for a 512-bit key, which is no outlier amongst post-quantum secure algorithm key sizes.

### RIJNDAEL inheritance

The original RIJNDAEL algorithm includes a block size option of 256 bits, which was not admitted for the AES standard. We decided to keep the 128 bits for EAES in order to ensure compatibility with existing hardware implementations<sup>1</sup>. Several mentioned attacks take advantage of the fact that AES-256 runs with the same block size as AES-128. Those vulnerabilities are at least mitigated in our algorithm by the higher number of rounds.

We also kept the 4 basic RIJNDAEL encryption functions and the RIJNDAEL algorithm architecture.

### More rounds and a 512-bit key version

We increased the number of rounds taking in account recommendations of renowned cryptographers

and cryptanalysts such as Bruce Schneier. The 256-bit key variant EAES-256 runs 22 rounds of the original RIJNDAEL transformation function, which is 8 more rounds than AES-256, and twice the best known attack which breaks 11 rounds [4]. Based on the same considerations, we fixed the number of 30 rounds for EAES-512.

### Say goodbye to 128- and 192-bit keys

Organizations such as the NIST and the EU quantum flagship [11] recommend to use the maximal key lengths of currently used cryptographic algorithms whenever possible. But for operational staff in IT, the options very often simply come down to the question about what choices they have for establishing a confidential communication with a business or cooperation partner. EAES doesn't offer the 128- and 192-bit key options anymore.

### Replacing *ExpandKey*

The invertibility of the original RIJNDAEL key schedule *ExpandKey* opened the door for various published attack schemes. Furthermore the output of the original RIJNDAEL key schedule does not offer cryptographic quality. In the simple cases, round keys could be extracted within encryption or decryption processes, and the original key computed by applying the inverted key schedule function. For cryptanalysis attacks, this is just another convenient property for finding a replacement of the chain of RIJNDAEL functions.

Therefore we replace *ExpandKey* by cryptographically secure Pseudo Random Number Generators (PRNGs) with strong diffusion properties in EAES.

### Pseudo Random Number Generators

PRNGs are built for deriving keys of a fixed size for further cryptographic operations by using an underlying pseudo random function. In our case those pseudo random functions are hashing algorithms. They are not even injective, but produce a well defined and collision resistant output.

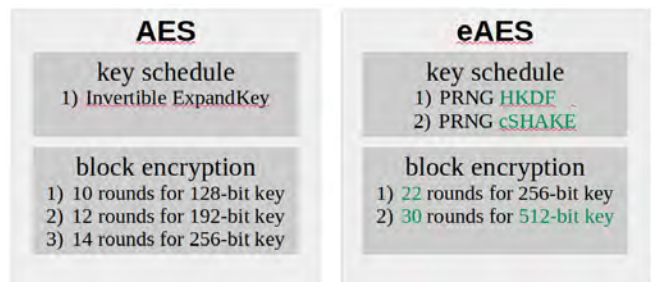


Figure 2: High level comparison of AES and eAES.

Furthermore, it is very cost-intensive under foreseeable technical developments to perform brute-force attacks on the chosen PRNGs. So they are also hardly reversible apart from the non-existent mathematical invertibility.

<sup>1</sup>We have also implemented an authenticated stream cipher (RCS) using the 256-bit block transform along with a cryptographically strong key-schedule, and an increase in transformation rounds that parallels eAES.



## Key schedule variants

Our first choice is HKDF, a HMAC-based extract-and-expand key derivation function, with SHA-2 as the underlying hash function. As a second option, we implemented the KECCAK-derived and customizable CSHAKE-function. KECCAK, the SHA-3 competition finalist, was chosen by the NIST for its algorithmic unrelatedness from SHA-2, while offering flexibility and a comparable speed in computation.

### Common features

Both HKDF and CSHAKE produce arrays of bytes which are converted to big-endian-ordered 32-bit integers for the round subkeys. Each round key has the same size as the cipher's block size, namely 128 bits. So the output of our key derivation functions needs to be of the size  $(r + 1) \cdot 128$  bit, where  $r$  represents the number of rounds. We renounce on the usage of a salt due to the fact that within our algorithm, the input is cryptographically strong already.

### HKDF

We consider the HMAC-based HKDF as a sensible intermediary solution for the derivation of the round keys, because it is already widely available. We use HKDF(SHA-256) for EAES-256 and HKDF(SHA-512) for EAES-512 to align with expected security strengths.

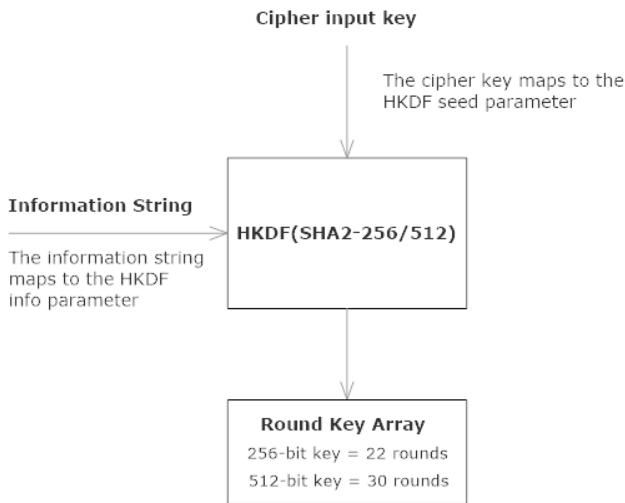


Figure 3: Rijndael HKDF eXtension.

SHA-256 and SHA-512 are members of the SHA-2 function family. They use the Merkle–Damgård construction, a method of building collision-resistant cryptographic hash functions from collision-resistant one-way compression functions. The compression function for SHA-2 uses the Davies–Meyer structure from a (classified) specialized block cipher.

HKDF generates cryptographically strong output of any desired length by repeatedly generating hash-blocks, concatenating them, and finally truncating the result to the desired length. Each call to HMAC involves two

calls to the SHA-2 hash function to generate a pseudo-random 256-bit or 512-bit output block.

So if the number of rounds is  $r$  and the SHA-2 output length is  $n$ , SHA-2 has to be called  $\lceil \frac{(r+1) \cdot 128}{n} \rceil$  times. This results in 12 times with SHA-256 for EAES-256 and 8 times with SHA-512 for EAES-512.

### CSHAKE

The SHA-3 competition finalist KECCAK is a so called *sponge function*. Those are functions with finite internal state, taking an input bit stream of any length and producing an output bit stream of any desired length.

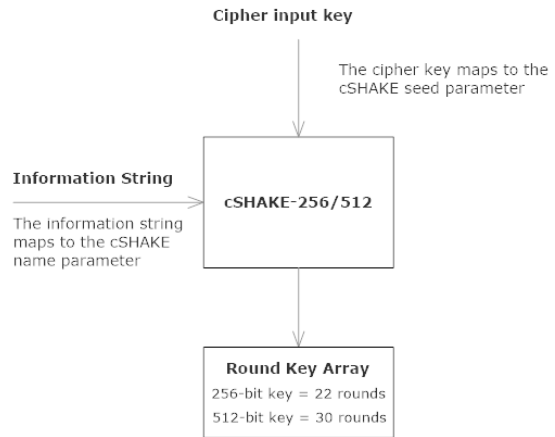


Figure 4: Rijndael SHAKE eXtension.

We implemented the SHA-3-derived SHAKE-function as a second option for the key schedule. CSHAKE is the customizable version of SHAKE. It is originally designed for 128- and 256-bit security strength [12]. Underhill has additionally created a 512-bit security implementation, which mirrors the internal block-size, squeeze and permutation settings of SHA3-512. CSHAKE-256 is used for EAES-256, and CSHAKE-512 is used for EAES-512.

The initial state of SHAKE is a  $5 \times 5$  array of 64-bit unsigned integers, 1600 zero bits it total. The first call is to initialize the custom state. Then the original EAES-key is absorbed into the previously initialized state, and with each call to the inner permutation function of KECCAK, rates of  $(1600 - 2 \cdot n, n \in \{256, 512\})$  bits are returned. So CSHAKE-256 returns 136, and CSHAKE-512 returns 72 pseudo-random bytes per call.

So if the number of EAES-rounds is  $r$  and the number of returned bytes per rate are  $n$ , we have  $\lceil \frac{(r+1) \cdot 128}{n \cdot 8} \rceil$  calls to the inner permutation functions of CSHAKE. For EAES-256 we have 3 and for EAES-512 we have 7 calls to the inner permutation function, +1 call for the initial state.

Note that here we are talking about calls to the inner permutation function of KECCAK, while we are talking about calls to SHA-2 itself in the previous section. Fewer calls of the permutation function lead to higher

efficiency of cSHAKE compared to the one of HKDF. On the other hand, KECCAK is prone to quantum algebraic attacks [7].

## Impact of Grover's search algorithm

Grover's algorithm offers a  $\sqrt{k}$  speed-up [8] over a classical exhaustive search (brute-force attack) on the set of keys of size  $k$ . It seems to be one of the most relevant quantum cryptanalytic impact for the study of block ciphers. The authors of [6] present quantum circuits to implement the key search for AES and analyze the quantum resources required to carry out such an attack for key sizes of 128, 192, and 256 bits.

The number of required logical qubits is relatively low, 6,681 for AES-256. On the other hand, the large circuit depth of enrolling the entire Grover iteration poses a challenge to an implementation on an actual physical quantum computer, even if the gates (basic quantum circuit operating on qubits) are not error-corrected. The key schedule *ExpandKey* causes much of the circuit cost within each Grover iteration. Replacing it by our chosen PRNGs will even be considerably more cost-intensive.

Here we only summarize the basic prerequisites of the involved procedures described in [6] for AES-256 and compare them to EAES for  $k \in \{256, 512\}$ .

### Algorithm architecture

A quantum circuit implementing a Boolean function  $f$  is the input of Grover's search algorithm:

$$f : \{0, 1\}^k \rightarrow \{0, 1\}$$

The basic algorithm finds an element  $x_0$  such that  $f(x_0) = 1$ . This is realized by repeatedly applying the operation  $G$  to measure an element  $x_0$  such that  $f(x_0) = 1$  with constant probability.

$$G = U_f \left( (H^{\otimes k} (2|0\rangle\langle 0| - 1_{2^k}) H^{\otimes k}) \otimes 1_2 \right)$$

$H$  denotes the  $2 \times 2$  Hadamard transform and  $U_f$  involves the computation of the cipher functions. To ensure uniqueness of the solution (the found key), a small number of plaintext–ciphertext pairs are needed. This number is 5 for a 256-bit key and 9 for 512-bit key.

### Quantum resources and graph theory

Quantum resources are represented by logical qubits, gates, and circuit depth. In the model of a Boolean circuit as a directed acyclic graph, the circuit depth is the maximal length of a path from an input gate to the output gate. The sum of required gates represent the complexity of an algorithm.

Reversible circuits are needed for the application of Grover on AES to provide invertibility of the operations. Therefore the proposed solution in [6] relies on a set

of fault-tolerant reversible logical gates. It consists of called Toffoli gates, controlled NOT gates and NOT gates. A Toffoli gate is a universal reversible logic gate, i.e. any reversible circuit can be constructed from Toffoli gates.

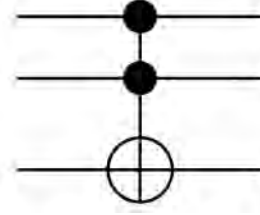


Figure 5: Toffoli gate (controlled-controlled-not gate).

For our key schedule replacements, we will base our comparisons on the results of “Estimating the cost of generic quantum pre-image attacks on SHA-2 and SHA-3s” [13]. The authors present an implementation including Grover's algorithm on reversible circuits on a surface-code-based fault-tolerant quantum computer.

### Quantum resources for the search

$H$  represents the Hadamard operation on a single qubit. A single solution to the equality function  $f$  can be found by applying  $H^{\otimes k+1}$  to the initial state, where  $k$  is the size of the key. The next step is applying

$$G^{\lceil \frac{\pi}{4} \sqrt{2^k} \rceil},$$

followed by a measurement of the entire quantum register which will yield a solution  $x_0$  with high probability.

The exact decomposition of the search is outlined in [6], Sect. 2. For our comparison we only look at the cost for the operation  $(2|0\rangle\langle 0| - 1_{2^k})$ , which is determined by the reduction to the implementation of a  $k$ -fold NOT gate, where  $k \in \{256, 512\}$  is the key size. In terms of Toffoli gates we have the formula  $8k - 24$  [14] and if we count only T-gates, we have the formula  $32k - 84$  as an upper bound for a  $k$ -fold controlled NOT gate [15]. This results in:

#### Cost for the operation $(2|0\rangle\langle 0| - 1_{2^k})$ :

key size	Toffoli gates	T-gates
256	2,024	8,108
512	4,072	16,300

Grover's search will have to be performed on the small number  $n$  of plaintext–ciphertext pairs. The equality function  $f$  can be implemented by a multiple controlled NOT gate that has  $128 \cdot n$  controls. With above formulas, the needed number of Toffoli counts and T-counts to compute the equality function  $f$  come down to:

#### Cost for the equality function $f$ :

key size	Toffoli counts	T-counts
256	5,096	20,396
512	9,192	36,780

The search resources only depend on the key sizes and not on the ciphers and their options. In this regard we don't have to include further differing considerations.

### Quantum resources for the ciphers

The number of rounds depends on the specific key length  $k$ , while the four basic RIJNDAEL functions are independent of the key length  $k$  for both AES and EAES. The realization of all RIJNDAEL functions is analyzed in dedicated sections in [6]. Here we only present the results and comparisons per function.

#### Cost for SHA-2 and -3 functions (256 bit):

function	gates	depth	qubits
SHA-256	$1.42 \cdot 2^{146}$	$1.69 \cdot 2^{144}$	$2^{12.6}$
SHA3-256	$1.52 \cdot 2^{147}$	$1.33 \cdot 2^{137}$	$2^{20}$

The values for SHA-256 and SHA3-256 are results from the resource estimates done in [13], converted to a representation in powers of 2 instead of 10. For both functions, the total cost comes down to approximately  $2^{166}$  basic operations. The estimation of the resources for the according 512-bit versions will be done in a further step of our research.

For our EAES-options we have to multiply the gate values by the number of calls to the hash and inner permutation functions.

#### Cost for AES and EAES key expansion (256 bit):

function	gates	depth	qubits
<i>ExpandKey</i>	54,331	23,896	512
HKDF	$1.07 \cdot 2^{150}$	$1.26 \cdot 2^{148}$	$2^{12.6} + 3072$
CSHAKE	$1.52 \cdot 2^{149}$	$1.33 \cdot 2^{139}$	$2^{20} + 1024$

The numbers of gates and depths take the number of calls to the according hash and permutation functions into account. Additional qubits are needed to store the output of all calls to the hash functions.

As mentioned before, we cannot refer to values for 512-bit keys at this time. But we can see that the 256-bit versions already offer considerable advantages over the original RIJNDAEL key schedule *ExpandKey*.

#### Cost for holding the state and initial round:

function	gates	depth	qubits
state	0	0	128
initial round $v_1$	64 NOT	0	128
initial round $v_2$	128 CNOT	1	256

Initial round  $v_1$  represents the realization with flipping bits. These values are the same for all ciphers and options.

#### Cost for basic functions

function	gates	depth	qubits
<i>SubBytes</i>	22,326	1	9
<i>ShiftRows</i>	0	0	0
<i>MixColumns</i>	277 CNOT	39	0
<i>AddRoundKey</i>	128 CNOT	1	0

The current round key is available on 128 dedicated wires. *SubBytes* is realized by finding the inverse of the byte in  $GF(2^8)$ , seen as permutation in  $\mathbb{F}_{256}$ . No extra gates are necessary to implement *ShiftRows*, as it corresponds to a permutation of the qubits. The position of the subsequent gates is simply adjusted to the correct input wire.

#### Cost for RIJNDAEL rounds

To compute all rounds of RIJNDAEL, 536 qubits are needed for 10 rounds in the 128-bit key version and 664 qubits are required for any round number  $r \geq 12$ . In this regard, the higher number of rounds in EAES does not add complexity to the algorithm. It does add complexity to the number of gates and to the depth, though.

#### Cipher output for 256-bit keys

The number of gates and depth for EAES is derived as follows:

Values from the rounds row in table 4, AES-256 in [6] :

$$g_{14} = \text{sum of number of gates in [6]}$$

$$d_{14} = \text{sum of depths in [6]}$$

Values from the key expansion cost table in this article:

$$g_{hkdf} = \text{number of gates for HKDF-256}$$

$$d_{hkdf} = \text{depths for HKDF-256}$$

$$q_{hkdf} = \text{number of qubits for HKDF-256}$$

$$g_{shake} = \text{number of gates for CSHAKE-256}$$

$$d_{shake} = \text{depths for CSHAKE-256}$$

$$q_{shake} = \text{number of qubits for CSHAKE-256}$$

Then we compute the values for EAES(HKDF) from the formulas:

$$gates = g_{hkdf} + \frac{22}{14}g_{14}, \quad depth = d_{hkdf} + \frac{22}{14}d_{14}$$

The values for EAES(CSHAKE) from the formulas:

$$gates = g_{shake} + \frac{22}{14}g_{14}, \quad depth = d_{shake} + \frac{22}{14}d_{14}$$

And the number of qubits for is the simple addition  $664 + q_{sha}$  ,  $664 + q_{sha3}$  respectively.

### Cost for each cipher output (256 bit)

cipher	gates	depth	qubits
AES-256	$1.65 \cdot 2^{21}$	$1.46 \cdot 2^{17}$	1,336
EAES <sub>(HKDF)</sub>	$1.07 \cdot 2^{150}$	$1.26 \cdot 2^{148}$	$1.21 \cdot 2^{13}$
EAES <sub>(CSHAKE)</sub>	$1.52 \cdot 2^{149}$	$1.33 \cdot 2^{139}$	$1.002 \cdot 2^{20}$

Note that the gates and depth values for EAES equal the values in the key expansion cost table. This comes from the fact that the contribution of the gates and depth from the rounds is in much lower power of 2 than the cost for the key expansions HKDF and CSHAKE. So we see that this replacements are the essential change and contribution to a higher algorithm complexity.

### Grover algorithm

For the overall T-count for Grover on RIJNDAEL we have an estimate of  $\lfloor \frac{\pi}{4} 2^{k/2} \rfloor \cdot (6t_k + f_k)$ , where  $k$  is the key size,  $t_k$  is the number of T-gates and  $f_k$  is the cost for the equality function. For the overall circuit depth we obtain with  $r$  = number of rounds:  $2 \cdot r \cdot$  total T-depth. Here we are ignoring some of the gates which do not contribute significantly to the bottom line. Both formulas are from [6], Sect. 3.4. So by a moderate estimation we get the following complexities for Grover's search on the compared ciphers:

### Grover's search on ciphers

cipher	gates	depth	qubits
AES-256	$3.24 \cdot 2^{151}$	$1.71 \cdot 2^{145}$	6,681
EAES <sub>(HKDF)</sub>	$> 2^{278}$	$> 2^{274}$	$> 2^{14}$
EAES <sub>(CSHAKE)</sub>	$> 2^{277}$	$> 2^{257}$	$> 2^{21}$

Both HKDF and CSHAKE have a considerable impact on the increased cost of Grover's search. The higher number of rounds for EAES-256 and the 512-bit key version additionally contribute to the complexity of the rounds by factors of at least  $r/14$ . The authors of [6] recommended in 2015 to move away from AES-128 when expecting the availability of at least a moderate-size quantum computer. Our implementation excludes that option and offers an increased complexity compared to AES-256.

### Impact of quantum algebraic attacks

The authors of [7] present an algorithm which leads to new considerations of the security of systems which can be reduced to solving Boolean equations. A solution  $a$  for the equation  $\mathcal{F} \cdot a = 0$  with a set of polynomials  $\mathcal{F} \subset \mathbb{C}[X]$  is called *Boolean* if each coordinate of  $a$  is either 0 or 1.

The resulting quantum algebraic attack algorithm includes quantum-monomial solving of polynomial systems over  $\mathbb{C}$  by applying a Macaulay linear system. Like this they constructed a Boolean equation solving algorithm, with the following properties:

- 1) It decides if there exists a Boolean solution.
- 2) It returns a Boolean solution with a given probability if there are such solutions to the system.
- 3) It returns  $\emptyset$  if no Boolean solution exists.

The runtime complexity of the resulting quantum algebraic attack is considerably lower than the one of Grover's Search, but it depends on two factors: a constant  $c$  and a condition number  $\kappa$ . The complexity of  $2^{78.53} c \kappa^2$  for AES-256 is not much higher than the complexity of  $2^{73.30} c \kappa^2$  for AES-128 due to the same block size of 128 bit. Therefore we can assume that the complexity won't be much higher for 512-bit key sizes, and it will also not considerably increased by the higher number of rounds in EAES.

The conclusion of [7] is that systems which can be solved by Boolean equation solving are only secure under quantum algebraic attack if the condition number  $\kappa$  is large. The construction of such systems is a topic for further research. Besides AES and KECCAK, stream ciphers such as TRIVIUM and the multivariate public key cryptosystem MPKC are affected by the attack.

### Conclusion

Considering the lower exponent of the highest power of 2 in the total algorithm complexity, EAES offers a higher complexity of a factor  $\geq 2^{277-151} = 2^{126}$  regarding Grover's search algorithm compared to AES, even if only the 256-bit version is used. Regarding the quantum algebraic attacks, we can say that there is no attack outlined yet for the version with HKDF. Regarding the version with CSHAKE, the quantum algebraic attack has to be adapted to two phases: KECCAK and then on the rounds functions. The complexity of such a solution remains to be investigated.

We consider EAES a sensible candidate for a first-generation post-quantum secure symmetric encryption. It runs efficiently on currently used devices and is compatible with existing hardware implementations. We hope it is able to contribute to a smooth transition into the new post-quantum cryptographical era.



## References

- [1] S. Kovac and J. Underhill, Towards post-quantum symmetric cryptography, <https://eprint.iacr.org/2019/553>.
- [2] J. Underhill, The CEX Cryptographic library in C++, <https://github.com/Steppenwolfe65/CEX>.
- [3] A. Biryukov and D. Khovratovich, Related-key Cryptanalysis of the Full AES-192 and AES-256, <https://eprint.iacr.org/2009/317.pdf>.
- [4] A. Biryukov, O. Dunkelman, N. Keller, D. Khovratovich and A. Shamir Key Recovery Attacks of Practical Complexity on AES-256 Variants With Up To 10 Rounds <http://www.wisdom.weizmann.ac.il/~orrd/crypt/PracticalAES256.pdf>.
- [5] S. Lucks, Attacking Seven Rounds of RIJNDAEL under 192-bit and 256-bit Keys, <https://madoc.bib.uni-mannheim.de/10615>, 2000.
- [6] M. Grassl, B. Langenberg, M. Roetteler, R. Steinwand, Applying Grover's algorithm to AES: quantum resource estimates, <https://arxiv.org/abs/1712.06239>, 2018.
- [7] Y.-A. Chen, X.-S. Gao, Quantum Algorithms for Boolean Equation Solving and Quantum Algebraic Attack on Cryptosystems, <https://arxiv.org/abs/1712.06239>, 2018.
- [8] L. K. Grover, A fast quantum mechanical algorithm for database search, Gary L. Miller, editor, *Proceedings of the Twenty-Eighth Annual ACM Symposium on the Theory of Computing (STOC 1996)*, pages 212–219 ACM, 1996.
- [9] Federal Information Processing Standards Publication 197, NIST, Announcing the ADVANCED ENCRYPTION STANDARD (AES), <https://csrc.nist.gov/csrc/media/publications/fips/197/final/documents/fips-197.pdf>, 2000.
- [10] C. Cid, Information Security Group, University of London, Algebraic Analysis of AES, <https://www.cosic.esat.kuleuven.be/ecrypt/AESday/slides/AES-Day-CarlosCid.pdf>, October 2012.
- [11] J. Baloo, KPN, Everything is Quantum – The EU Quantum Flagship, <https://2017.pqcrypto.org/conference/slides/baloo.pdf>, 2017.
- [12] J. Kelsey, S. -J. Chang, R. Perlner, SHA-3 Derived Functions: cSHAKE, KMAC, TupleHash and ParallelHash, <https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-185.pdf>, December 2016.
- [13] M. Amy, O. Di Matteo, V. Gheorghiu, M. Mosca, A. Parent, J. Schanck, Estimating the cost of generic quantum pre-image attacks on SHA-2 and SHA-3s, <https://eprint.iacr.org/2016/992.pdf> QCrypt, 2016.
- [14] A. Barenco, C.H. Bennett, R. Cleve, D.P. DiVincenzo, N. Margolus, P.W. Shor, T. Sleator, J. Smolin, H. Weinfurter, Elementary gates for quantum computation, *Phys. Rev. A*, 52(5):3457–3467, 1995.
- [15] N. Wiebe, M. Roetteler, Quantum arithmetic and numerical analysis using Repeat-Until-Success circuits, <https://arxiv.org/abs/1406.2040>, 2014.

## Eine Gerade dreht sich um eine andere: Die Kühlturmfläche

**J. Meyer**  
(Hameln)

j.m.meyer@t-online.de



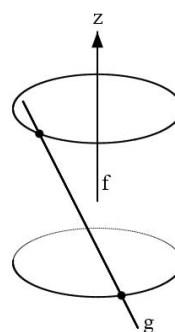

---

### Einführung

---

Was für eine Fläche entsteht, wenn eine Gerade  $g$  um eine feste andere Gerade  $f$  rotiert? Sind  $g$  und  $f$  zueinander parallel, entsteht ein *Zylinder*. Falls sich beide Geraden schneiden, bekommt man einen *Doppelkegel*. Interessanter ist es, wenn beide Geraden zueinander windschief sind: Dann entsteht ein *einschaliges Hyperboloid*. Dieses wird auch hinsichtlich seiner Tangenten und Schnitte untersucht.

Im Anhang wird für quadratische Flächen eine Lösung des Hidden-Line-Problems vorgestellt.



**Abbildung 1:**  $g$  rotiert um  $f$

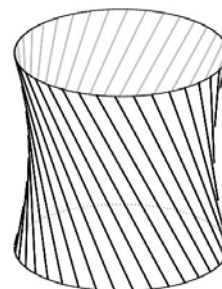
Wie bekommt man die Gleichung der von  $g$  überstrichenen Fläche? Schreibt man  $d := \frac{1 - \cos \varphi}{2}$ , so ist einerseits (mit CAS-Hilfe)

$$x^2 + y^2 - 1 = 4 \cdot d \cdot (t - 1) \cdot t$$

und andererseits  $t = \frac{1-z}{2}$ ; zusammen folgt

$$x^2 + y^2 - d \cdot z^2 = 1 - d.$$

Ist  $\varphi = 0^\circ$ , also  $d = 0$ , so bekommt man einen *Zylinder* mit  $x^2 + y^2 = 1$  ( $z$  beliebig). Ist  $\varphi = 180^\circ$ , also  $d = 1$ , so lautet die Flächengleichung  $x^2 + y^2 = z^2$ ; man hat also einen *Doppelkegel*. Diese beiden trivialen Fälle seien für die weiteren Betrachtungen ausgeschlossen. Dann ist  $0 < d < 1$ , und die beiden Geraden  $g$  und  $f$  sind zueinander windschief (Abb. 2).



**Abbildung 2:** Um eine Gerade rotiert eine dazu windschiefe Gerade

---

### Vorbemerkung

---

Im Folgenden wird mit Punkten gerechnet wie mit den zugehörigen Ortsvektoren.

---

### Das einschalige Hyperboloid und seine Geraden

---

Die feste Gerade  $f$  sei die  $z$ -Achse, und die rotierende Gerade  $g$  verlaufe durch die Punkte

$$\begin{pmatrix} \cos \alpha \\ \sin \alpha \\ 1 \end{pmatrix} \quad \text{und} \quad \begin{pmatrix} \cos(\alpha + \varphi) \\ \sin(\alpha + \varphi) \\ -1 \end{pmatrix},$$

sie hat also den allgemeinen Punkt

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} \cos \alpha \\ \sin \alpha \\ 1 \end{pmatrix} + t \cdot \begin{pmatrix} \cos(\alpha + \varphi) - \cos \alpha \\ \sin(\alpha + \varphi) - \sin \alpha \\ -2 \end{pmatrix}.$$

## Geraden auf der Fläche und Tangenten

Zunächst fällt auf: Da  $\cos \varphi = \cos(-\varphi)$  ist, überstreichen die Geraden durch die Punkte

$$\begin{pmatrix} \cos \alpha \\ \sin \alpha \\ 1 \end{pmatrix} \quad \text{und} \quad \begin{pmatrix} \cos(\alpha - \varphi) \\ \sin(\alpha - \varphi) \\ -1 \end{pmatrix}$$

dieselbe Fläche; diese enthält demnach (mindestens) zwei Scharen von Geraden (Abb. 3). Man findet solche Flächen bei den Ummantelungen von Kühltürmen; die beiden Geradenscharen verleihen solchen Bauwerken eine hohe Stabilität.

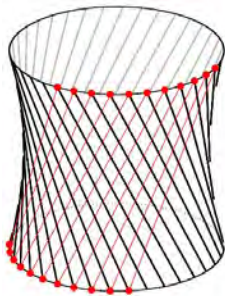


Abbildung 3: Zwei Geradenscharen auf der Fläche

Vielleicht gibt es noch mehr Geraden auf der Fläche? Und was lässt sich über Tangenten der Fläche sagen? Beginnen wir mit der zweiten Frage. Sei dazu

$$P = \begin{pmatrix} x_0 \\ y_0 \\ z_0 \end{pmatrix}$$

ein beliebiger Punkt der Fläche und

$$P + s \cdot R$$

mit Richtungsvektor

$$R = \begin{pmatrix} u \\ v \\ w \end{pmatrix}$$

ein allgemeiner Punkt auf einer Geraden durch  $P$ . Um herauszufinden, für welche Richtungsvektoren  $R$  diese Gerade Tangente ist, müssen wir sie mit der Fläche schneiden; die Schnittgleichung

$$s(s(u^2 + v^2 - dw^2) + 2(ux_0 + vy_0 - dwz_0)) = 0$$

muss dann  $s = 0$  als doppelte Nullstelle haben. Dies führt auf

$$\begin{pmatrix} u \\ v \\ w \end{pmatrix} \cdot \begin{pmatrix} x_0 \\ y_0 \\ -d \cdot z_0 \end{pmatrix} = 0;$$

alle Tangenten durch

$$P = \begin{pmatrix} x_0 \\ y_0 \\ z_0 \end{pmatrix}$$

stehen somit auf

$$\begin{pmatrix} x_0 \\ y_0 \\ -d \cdot z_0 \end{pmatrix}$$

senkrecht und bilden mithin eine Ebene. Diese *Tangentialebene* hat die Gleichung

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} \cdot \begin{pmatrix} x_0 \\ y_0 \\ -d \cdot z_0 \end{pmatrix} = \begin{pmatrix} x_0 \\ y_0 \\ z_0 \end{pmatrix} \cdot \begin{pmatrix} x_0 \\ y_0 \\ -d \cdot z_0 \end{pmatrix} = 1 - d$$

und durchsetzt die Fläche.

Wenn die Gerade mit dem allgemeinen Punkt

$$P + s \cdot R$$

(mit  $P = \begin{pmatrix} x_0 \\ y_0 \\ z_0 \end{pmatrix}$  als beliebigem Punkt der Fläche und

$R = \begin{pmatrix} u \\ v \\ w \end{pmatrix}$  als Richtungsvektor) *vollständig* in der

Fläche verlaufen soll, muss die obige Schnittgleichung für *alle*  $s$  erfüllt sein, d.h. es muss nicht nur

$$\begin{pmatrix} u \\ v \\ w \end{pmatrix} \cdot \begin{pmatrix} x_0 \\ y_0 \\ -d \cdot z_0 \end{pmatrix} = 0$$

gelten, sondern außerdem noch

$$u^2 + v^2 = d \cdot w^2.$$

Nun kann keine der ganz in der Fläche verlaufenden Geraden parallel zur  $x$ - $y$ -Ebene sein, so dass man ohne Bedenken  $w = 1$  setzen kann. Aufgrund der Rotationssymmetrie der Fläche kann man  $x_0 = 0$  annehmen; dann ist  $y_0 \neq 0$ . Die beiden entstehenden Gleichungen führen zur Lösung

$$u = \pm \frac{\sqrt{d} \cdot \sqrt{y_0^2 - d \cdot z_0^2}}{y_0}.$$

Aufgrund der Flächengleichung ist

$$y_0^2 - d \cdot z_0^2 = 1 - d \in (0; 1);$$

somit gibt es für  $u$  stets genau zwei Lösungen. Es gibt demnach nur die beiden schon bekannten Geradenscharen mit dem allgemeinen Punkt

$$\begin{pmatrix} 0 \\ y_0 \\ z_0 \end{pmatrix} + \frac{s}{y_0} \cdot \begin{pmatrix} \pm \sqrt{d \cdot (1 - d)} \\ d \cdot z_0 \\ y_0 \end{pmatrix}$$

auf der Fläche. Natürlich liegen sie auch in der Tangentialebene zu  $\begin{pmatrix} 0 \\ y_0 \\ z_0 \end{pmatrix}$ .

## Schnitte

Schneidet man die Fläche mit der Ebene zu  $z = z_0$ , bekommt man natürlich einen Kreis. Schneidet man mit der Ebene zu  $x = x_0$ , so ergibt sich

$$y^2 - d \cdot z^2 = 1 - d - x_0^2.$$

Für  $x_0^2 = 1 - d$  bekommt man *Geraden*, nämlich die mit den allgemeinen Punkten

$$\begin{pmatrix} \sqrt{1-d} \\ \pm \sqrt{d} \cdot z \\ z \end{pmatrix} = \begin{pmatrix} \sqrt{1-d} \\ 0 \\ 0 \end{pmatrix} + z \cdot \begin{pmatrix} 0 \\ \pm \sqrt{d} \\ 1 \end{pmatrix}$$

sowie

$$\begin{pmatrix} -\sqrt{1-d} \\ \pm \sqrt{d} \cdot z \\ z \end{pmatrix} = \begin{pmatrix} -\sqrt{1-d} \\ 0 \\ 0 \end{pmatrix} + z \cdot \begin{pmatrix} 0 \\ \pm \sqrt{d} \\ 1 \end{pmatrix}.$$

Diese Geraden müssen in den oben schon erwähnten Geradenscharen enthalten sein. Ein direkter Nachweis ist unnötig.

Ist  $x_0^2 < 1 - d$ , so bekommt man

$$y^2 - d \cdot z^2 = 1 - d - x_0^2 > 0$$

und somit eine seitlich geöffnete *Hyperbel* (Abb. 4).

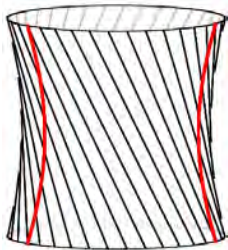


Abbildung 4: Hyperbel als Schnittkurve

Aus diesem Grunde heißt die Fläche (*einschaliges Hyperboloid*).

Ist  $x_0^2 > 1 - d$ , so bekommt man

$$y^2 - d \cdot z^2 = 1 - d - x_0^2 < 0$$

und somit eine nach oben und unten geöffnete *Hyperbel* (Abb. 5).

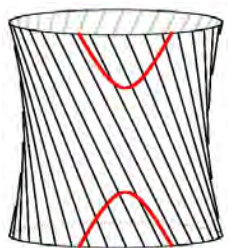


Abbildung 5: Eine andere Hyperbel als Schnittkurve

Schneidet man das Hyperboloid mit Ebenen, die nicht zu den Koordinatenebenen parallel sind, erhält man auch Ellipsen und Parabeln.

## Anhang: Eine einfache Lösung des Hidden-Line-Problems

Bei der Erstellung der Abbildungen war das Hidden-Line-Problem zu lösen, da einige Geradenteile von anderen überdeckt wurden. Da die Flächengleichung nur quadratisch ist, bietet sich folgende Lösung an, die auch wieder vom Einsatz eines CAS profitiert: Vorausgesetzt wird, dass es sich bei den Abbildungen um das Resultat einer *geraden Parallelprojektion* handelt; das räumliche Objekt wird auf die Ebene mit

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} \cdot N = 0$$

abgebildet. Zu jedem Flächenpunkt

$$P = \begin{pmatrix} x_0 \\ y_0 \\ z_0 \end{pmatrix}$$

gibt es einen zum Normalenvektor

$$N = \begin{pmatrix} n_x \\ n_y \\ n_z \end{pmatrix}$$

parallelen Projektionsstrahl mit dem allgemeinen Punkt

$$X = P + \lambda \cdot N.$$

Dieser Strahl schneidet die Fläche außer in  $P$  noch in einem weiteren Punkt  $Q$ . Da die Flächengleichung quadratisch ist, lässt sich das zu  $Q$  gehörige  $\lambda \neq 0$  leicht ausrechnen: Es ist

$$\lambda_Q = -2 \cdot \frac{N \cdot \begin{pmatrix} x_0 \\ y_0 \\ -d \cdot z_0 \end{pmatrix}}{n_x^2 + n_y^2 - d \cdot n_z^2}.$$

Falls  $\lambda_Q$  positiv ist, liegt  $Q$  zwischen dem Auge und  $P$  (Abb. 6);  $P$  ist dann unsichtbar.

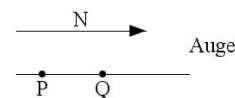


Abbildung 6:  $P$  ist unsichtbar

In der Formel für  $\lambda_Q$  erkennt man einiges an *Struktur*. So tritt der Normalenvektor der Tangentialebene zu  $P$  auf; außerdem erkennt man, dass die Form des Nenners auch in der Flächengleichung

$$x^2 + y^2 - d \cdot z^2 = 1 - d$$

erscheint. Ordnet man dem Punkt

$$X = \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

den Punkt

$$X^* := \begin{pmatrix} x_0 \\ y_0 \\ -d \cdot z_0 \end{pmatrix}$$

zu, so schreibt sich die Flächengleichung einfach als

$$X \cdot X^* = 1 - d$$

(mit dem gewöhnlichen Skalarprodukt), und die zu  $P$  gehörige Tangentialebene hat die Gleichung

$$X \cdot P^* = 1 - d.$$

Schneidet man den Projektionsstrahl, dessen allgemeiner Punkt

$$P + \lambda \cdot N$$

ist, mit dem Hyperboloid, bekommt man

$$\begin{aligned} & (P + \lambda \cdot N) \cdot (P^* + \lambda \cdot N^*) \\ &= P \cdot P^* + 2 \cdot \lambda \cdot N \cdot P^* + \lambda^2 \cdot N \cdot N^* \\ &= 1 - d, \end{aligned}$$

woraus sich die Lösungen

$$\lambda = 0$$

und

$$\lambda = -\frac{P \cdot N^* + P^* \cdot N}{N \cdot N^*}$$

sofort ergeben; man hat demnach das gleiche Resultat wie oben. Das ist mitunter so beim CAS-Einsatz: Wenn man das Ergebnis sieht, kommt man auch auf einem anderen Weg dorthin, der die ganze Rechnerei überflüssig macht.

Allerdings wird mit der Ungleichung  $\lambda_Q > 0$  zu viel als unsichtbar deklariert, denn in Wahrheit kann man auch oben in die Fläche hineingucken. Man muss daher sagen:  $P$  ist unsichtbar, wenn  $\lambda_Q > 0$  ist und wenn die  $z$ -Koordinate von  $Q$  im Bereich  $[-1; 1]$  liegt.

Selbstverständlich lässt sich die Hidden-Line-Problematik auch vollständig an ein CAS oder Zeichenprogramm delegieren; man sieht aber, dass man bei quadratischen Flächen auch mit einfacher Vektorgeometrie weiterkommt. Das hier vorgestellte Verfahren wird aufwändiger, wenn man auch noch die zusätzliche Darstellung von Tangential- oder Schnittebenen berücksichtigen will.



### SFB/TRR 195 Symbolic Tools in Mathematics and their Application (Part 4/5)

#### Random matrices, free probability theory and compact quantum groups

One of the five core areas of the SFB-TRR 195 *Symbolic Tools in Mathematics and their Application* consists in random matrices, free probability theory and compact quantum groups. A central theme is noncommutativity, i.e. we deal with algebras whose multiplication is not necessarily commutative. Recall that a random matrix is a finite square matrix whose entries are chosen at random. Letting the size of the matrix tend to infinity, we may employ tools from free probability theory in order to handle the limit behaviour. More generally, free probability theory deals with a special kind of “free independence” amongst abstract random variables of a highly noncommutative nature. Compact quantum groups in turn are more general objects than compact groups, providing a suitable notion of symmetry in noncommutative topological or measurable settings, amongst others for free probability. Whereas free probability and compact quantum groups have been initiated in the 1980s, random matrix theory has a much longer history, but became a central topic in mathematics only in the last twenty or so years.

Within the SFB-TRR 195, in a project of Hannah Markwig and Roland Speicher links between various modern variants of Hurwitz numbers (given by factorizations in the permutation groups, subject to special constraints) and random matrices are being explored. Those investigations were also supported quite a bit by the expertise from one of the other core areas, as Hurwitz numbers often have also some interpretations in terms of tropical geometry. In another direction, non-commutative rational functions are being studied, dealing amongst others with the question when two rational expressions in non commuting variables actually express the same object. Results have been obtained via presentations in the free skew field; the latter is a quite beautiful non-commutative algebraic object, whose basic theory was developed in deep work of Paul Cohn in the last decades of the last century, but which is still considered as too non-commutative to be useful by most algebra aficionados. We hope that this perception is going to change; in particular, as such non-commutative rational functions are canonical objects where random

matrices can be plugged in. One of the main outcomes of a project of Speicher was the implementation of concrete algorithms for the computation of eigenvalue distributions for such functions of random matrices.

A third project in this fifth core area, by the author, deals with computational approaches to compact quantum groups of a matrix type. We study a particular class of such quantum groups – the so called “easy” quantum groups – whose representation theory can be described using partitions of sets, in a generalized Schur-Weyl sense. We express these underlying combinatorial objects as well as their operations using finite words making them accessible for the computer. The major challenge was to find first examples of quantum groups which also involve linear combinations of such words rather than just the plain words, the technical issue being stability under certain operations. We managed to find a large number of them with the help of our computer experiments. In a second, theoretical step, we were then able to interpret these new quantum groups as subrepresentations of previously known “easy” ones, hence understanding better the representation theory of the latter ones via this detour to the linear combinations. Moreover, taking the converse perspective, we are studying superrepresentations of those well-known quantum groups using a somewhat inverse construction. We thus obtained new kinds of extensions of quantum groups by cyclic groups, and more generally a number of new products of quantum groups which may only exist in the quantum setting.

With these three projects, we explore new applications of computer algebra tools to seemingly unrelated fields. We use computer based experiments for forming our intuition or for finding new objects which are then interpreted by theoretical means. This strategy led to a number of new insights.

Moritz Weber (Saarbrücken)



# Computeralgebra in Zahlentheorie und Arithmetischer Geometrie

## Andreas-Stephan Elsenhans (Würzburg)

Dem Lehrstuhl Algebra am Institut für Mathematik an der Julius-Maximilians-Universität Würzburg gehören Peter Müller und Andreas-Stephan Elsenhans sowie Joachim König (aktuell KAIST, Daejeon, Südkorea), Florian Möller im Mittelbau sowie die Doktoranden Dominik Barth und Andreas Wenz an.

Die Hauptthemen der Forschung dieses Lehrstuhls sind Galois- und gruppentheoretische Methoden zur Untersuchung spezieller Zahl- und Funktionenkörper. Seit der Berufung von Andreas-Stephan Elsenhans hat sich dies um Anwendungen in der arithmetischen Geometrie erweitert.

Die Forschung in Zusammenarbeit mit Jürgen Klüners (Paderborn) betrifft die Berechnung von Teilkörpern und Galoisgruppen von Zahlkörpern sowie die Konstruktion und Klassifikation spezieller Zahlkörper.

Gemeinsam mit Jörg Jahnel (Siegen) wird an der Konstruktion von del Pezzo Flächen mit gegebener Galoismodulstruktur der Picardgruppe gearbeitet. Weiter-

hin werden die Endomorphismen der Kohomologie von K3-Flächen untersucht.

Die Konstruktion spezieller del Pezzo Flächen steht in enger Verbindung mit den Ergebnissen der algebraischen Geometrie. Hier stehen verschiedene klassische Beschreibungen der Modulräume (z.B. durch Invarianten) im Vordergrund.

Die Untersuchung von K3-Flächen basiert neben lokalen Methoden wie  $p$ -adischem Zählen von Punkten über endlichen Körpern und der Berechnung der Periodenintegrale auch auf klassischen geometrischen Betrachtungsweisen, wie Singularitäten und Faserungen in Kurven kleinen Geschlechts.

Weitere Untersuchungen betreffen die in den gewonnenen Daten auftretenden statistischen Verteilungen, wie beispielsweise der Anzahl der rationalen Punkte beschränkter Höhe auf einer Varietät.

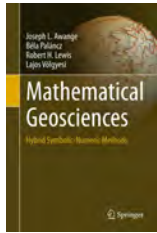
Implementierungen finden hauptsächlich in `magma`, `C` und `Python` statt.

---

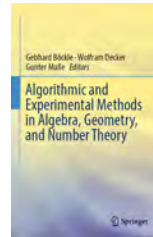
## Publikationen über Computeralgebra

---

### Neuerscheinungen:



Joseph L. Awange, Béla Paláncz,  
Robert H. Lewis, Lajos Völgyesi,  
*Mathematical Geosciences –  
Hybrid Symbolic-Numeric Methods*,  
Springer Int. Publishing, 2018,  
612 Seiten,  
ISBN 978-3-319-67370-7



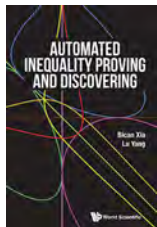
Gebhard Böckle, Wolfram Decker  
und Gunter Malle (Hrsg.),  
*Algorithmic and Experimental Me-  
thods in Algebra, Geometry, and  
Number Theory*,  
Springer Int. Publishing,  
2017, 763 Seiten,  
ISBN 978-3-319-70565-1



Tetsuo Ida,  
*Computational Origami*,  
Springer Int. Publishing, 2019,  
200 Seiten,  
ISBN 978-3-319-59188-9



Antonio Montes,  
*The Gröbner Cover*,  
Springer Nature, 2019,  
276 Seiten,  
ISBN 978-3-030-03903-5



Bican Xia, Lu Yang,  
*Automated Inequality Proving  
and Discovering*,  
World Scientific, 2017,  
334 Seiten,  
ISBN 978-9-814-75911-3

Die Rubrik Publikationen ist nicht allein auf eine Liste von Neuerscheinungen und Neuauflagen beschränkt. Sie lebt vor allem von fundierten Rezensionen von Fachgruppenmitgliedern für Fachgruppenmitglieder, die wir an dieser Stelle gerne abdrucken. Sollte eines der oben genannten Bücher, insbesondere eine der Neuerscheinungen, Ihr Interesse geweckt haben, und Sie möchten dieses für den Computeralgebra-Rundbrief besprechen, nehmen Sie bitte Kontakt zu Florian Heß oder Martin Kreuzer ([florian.hess@uni-oldenburg.de](mailto:florian.hess@uni-oldenburg.de), [martin.kreuzer@uni-passau.de](mailto:martin.kreuzer@uni-passau.de)) auf.

---

## Promotionen in der Computeralgebra

---

**Christian Kell:** A Structure-Based Attack on the Linearized Braid Group-Based Diffie-Hellman Conjugacy Problem in Combination with an Attack Using Polynomial Interpolation and the Chinese Remainder Theorem

Betreuer: Martin Kreuzer (Passau)

Zweitgutachter: Vladimir Shpilrain (New York)

Mai 2019

**Abstract:** Traditionelle Kryptosysteme basieren auf kommutativen algebraischen Strukturen. Zum Beispiel verwendet das RSA Kryptosystem den Ring  $\mathbb{Z}/n\mathbb{Z}$  für eine Zahl  $n$ , die ein Produkt zweier großer Primzahlen ist, und das El-Gamal Kryptosystem bedient sich der kommutativen Gruppe  $(\mathbb{Z}/p\mathbb{Z})^\times$  für eine große Primzahl  $p$ . Die ersten bedeutenden Vorschläge in der nicht-kommutativen Kryptographie waren der Schlüsselaustausch von Anshel-Anshel-Goldfeld bzw. das darauf aufbauende Kryptosystem von Ko et al.. Diese beruhen auf der Verwendung einer Plattformgruppe, in der das Wortproblem effizient lösbar ist, das Konjugatorsuchproblem aber nicht. Letzteres ist dabei das Problem, zu gegebenen Elementen  $a, b$  einer Gruppe  $G$ , von denen man weiß, dass sie konjugiert sind, den Konjugator  $c$  mit  $b = cac^{-1}$  zu finden.

Als erste und wichtigste Plattformgruppe wurde dabei die Zopfgruppe  $B_n$  verwendet, wobei man sich der Artin-Präsentation bediente, bzgl. der das Wortproblem effizient lösbar ist. Dieser erste Vorschlag wurde sogleich auf verschiedene Arten angegriffen:

- (1) V. Gebhardt und andere versuchten, die Konjugationsklassen in  $B_n$  möglichst geschickt zu durchsuchen

und produzierten längenbasierte *summit cycle* Angriffe, die bei kleineren  $n$  (bis etwa  $n = 80$ ) erfolgreich waren.

- (2) Cheon und Jun zeigten einen Angriff auf, der die Linearisierung von  $B_n$  mittels der Lawrence-Krammer Darstellung nutzte, um das Konjugatorsuchproblem auf ein lineares Gleichungssystem über einem endlichen (aber riesigen) Körper zurückzuführen und damit zu zeigen, dass es in polynomialer Zeit lösbar ist. Jedoch ist das sich ergebende Polynom von einem so hohen Grad, dass der Angriff nicht wirklich praktisch durchführbar ist.

Die vorliegende Dissertation befasst sich mit einer Optimierung des Angriffs von Cheon und Jun auf das Zopfgruppen-Kryptosystem von Ko et al. In einem ersten Schritt wird das entstehende lineare Gleichungssystem (LGS) über dem Laurent-Polynomring  $\mathbb{Z}_2[t, t^{-1}]$  strukturell analysiert und so weit wie möglich auf ein kleineres, dichteres LGS reduziert. Anschließend wird das reduzierte System auf Zeilenstufenform gebracht, was jedoch durch Interpolation auf eine endliche Menge von LGS über  $\mathbb{Z}$  zurückgeführt wird. Für jedes dieser LGS wiederum wird die Aufgabe durch Reduktion modulo verschiedener kleiner Primzahlen  $p$  und Rekombination der modularen Ergebnisse mit Hilfe des chinesischen Restsatzes auf eine endliche Menge von LGS über kleinen endlichen Körpern reduziert. Insgesamt ergeben sich eine Reduktion der Komplexität des Angriffs von mehreren Größenordnungen und ein Einblick in das Wachstumsverhalten der LGS und ihrer Lösungen bzgl. der Anzahl der Stränge  $n$  der Zopfgruppe  $B_n$ .

---

## Berufungen

---

**Prof. Dr. Max Horn** hat zum 1.4.2019 eine Professur für Algorithmische Algebra an der Universität Siegen angetreten.

**Prof. Dr. Ulrich Thiel** hat zum 1.8.2019 eine Professur für Algebra an der TU Kaiserslautern angenommen.

**Prof. Dr. Anne Frühbis-Krüger** hat zum 1.10.2019 eine Professur für Mathematik mit dem Schwerpunkt Arithmetische/Algebraische Geometrie und Computeralgebra an der Carl von Ossietzky Universität Oldenburg angetreten.

### MEGA 2019

Madrid, Spanien, 17.06. – 21.06.2019

[eventos.ucm.es/12097/detail/mega-2019.html](http://eventos.ucm.es/12097/detail/mega-2019.html)

Das nicht sonderlich gut googlebare Akronym MEGA steht für *Effektive Methoden in Algebraischer Geometrie*, und funktioniert am besten in diversen romanischen Sprachen. So auch in der Sprache des diesjährigen Austragungsortes Madrid. Die MEGA ist eine alle zwei Jahre an den verschiedensten Orten Europas stattfindende Konferenzreihe, die sich mit rechnerischen Aspekten und Anwendungen von algebraischer Geometrie beschäftigt, wobei „algebraische Geometrie“ in einem weiten Sinne zu verstehen ist. Aus der Konferenz entsteht in der Regel ein Sonderband des *Journals of Symbolic Computation*, wie auch in diesem Jahr. Auf der MEGA 2019 gab es 38 Vorträge jeweils in zwei parallelen Sektionen, die durch einen Begutachtungsprozess ausgesucht worden waren, dazu acht Plenarvorträge und außerdem Software-Präsentationen und Poster. Dabei ging es bei einer Menge der Vorträge tatsächlich um algebraische Geometrie im engeren Sinne, aber viele andere widmeten sich einer breiten Palette von Themen, die von reeller algebraischer Geometrie über Optimierung bis zur (Postquanten-) Kryptographie reichten. Mit etwa 130 Teilnehmern war die Konferenz gut besucht, und bot reichlich Gelegenheit zum Austausch am Rande der Vorträge, in den Kaffeepausen und beim gemeinsamen Mittagessen auf dem Gelände der Universidad Complutense de Madrid.

Das Organisationsteam um Mariemi Alonso hat eine so perfekte Arbeit geleistet, dass man von der Organisation fast nichts gespürt hat. Madrid erwies sich als ein ausgesprochen angenehmer Austragungsort. Trotz des weltstädtischen Charakters der Stadt konnte man leicht eine Unterkunft in fußläufiger Entfernung zum Tagungsort und zur Innenstadt wählen. Die nächste MEGA Konferenz wird vom 7. bis 11. Juni 2021 im norwegischen Tromsø, nördlich des Polarkreises, stattfinden.

Gregor Kemper (München)

### ISSAC 2019

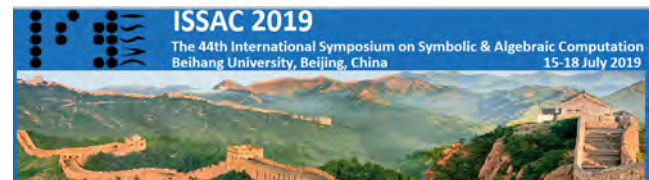
Peking, China, 15.07. – 18.07.2019

[www.issac-conference.org/2019](http://www.issac-conference.org/2019)

Das Akronym ISSAC steht *International Symposium on Symbolic and Algebraic Computation*, und damit für eine der bedeutendsten internationalen Konferenzen im Bereich Computeralgebra. Zwar hat die Fachgruppe Computeralgebra derzeit keinen Sitz im Steering Committee der ISSAC, ist jedoch weiter eng mit der ISSAC verbunden unter anderem durch die von uns vergebenen Preise. Mehr hierzu später.

Die ISSAC fand heuer an der Beihang University in Peking statt und war mit 148 Teilnehmern erneut gut besucht. Der General Chair war diesmal von dem Duo James Davenport und Dongming Wang besetzt, und die lokale Leitung lag in Händen von Chenqi Mou, dessen Hilfsbereitschaft und Organisationstalent einen wichtigen Beitrag zum guten Gelingen der Konferenz leisteten. Das Programmkomitee wurde von Manuel Kauers geleitet und wählte aus über 70 eingereichten Vorträgen 44 aus, die auf der ISSAC präsentiert wurden. Neben diesen — jeweils in zwei Sektionen abgehaltenen — eingereichten Vorträgen gab es drei eingeladene

Hauptvorträge und, als Auftakt zur Konferenz, drei Tutorials. Außerdem wurden 16 Poster und acht Software-Demos präsentiert, die auch einen Auswahlprozess durchlaufen hatten. Für weitere Details, Titel und Autoren sei auf die wie immer gut organisierte Homepage der Konferenz verwiesen.



*Issac 2019 in Peking*

Spannend war auch diesmal wieder das ISSAC Business Meeting, bei dem unter anderem von allen Teilnehmern der Konferenz der Austragungsort der ISSAC 2021 bestimmt wurde. Erfreulicherweise gab es drei Bewerbungen: aus Havanna (Kuba), Lille (Frankreich) und Sankt Petersburg (Russland). Von diesen setzte sich in der Wahl Sankt Petersburg durch. Außerdem war wieder ein neues Mitglied des Steering Committees zu wählen. Auch hierfür traten drei Kandidatinnen und Kandidaten an, von denen Shaoshi Chen (Chinese Academy of Sciences) gewählt wurde.

Nun zu den schon oben erwähnten Preisen, von denen auch auf dieser ISSAC eine ganze Reihe vergeben wurden: der Distinguished Paper Award, der Distinguished Student Author Award, der Distinguished Poster Award und der Distinguished Software Demonstration Award. Es sind die letzten beiden dieser Preise, an denen die Fachgruppe Computeralgebra als Sponsor des Preisgeldes von jeweils 250 Euro direkt beteiligt ist, weshalb in diesem Bericht auch nur diese beiden Preise weiter erwähnt werden sollen, während für die anderen auf die Konferenz-Homepage verwiesen sei. Erneut wurden für die Ermittlung der Preisträger spezielle Komitees aus den ohnehin bestehenden Poster- und Software-Komitees gebildet. Der Preis für das beste Poster ging an:

**Autoren:** Leili Rafiee Sevyeri und Robert Corless.

**Titel:** Approximate GCD in Bernstein basis.

Als beste Software-Demo wurde ausgezeichnet:

**Autoren:** Julián Cuevas-Rozo, Jose Divasón, Miguel Marco-Buzunáriz und Ana Romero.

**Titel:** A Kenzo Interface for Algebraic Topology Computations in SageMath.

Bei dem Bankett übten für manche westliche Konferenzteilnehmer sowohl die Speisen als auch die Aufführung von Szenen der Peking-Oper den Reiz des Ungewohnten aus. Auch etwas ungewohnt war die Herausforderung durch die „Great Firewall of China“, die viele gängige Internetseiten und -dienste blockiert. Mehr als ausgeglichen wurde dies jedoch durch die Freundlichkeit und Zugänglichkeit der Gastgeberstadt Peking.

Die nächste ISSAC-Konferenz findet vom 20. bis 23. Juli 2020 in Kalamata (Griechenland) statt.

Gregor Kemper (München)



## CASC 2019

Moskau, Russland, 26.08. – 30.08.2019

[www.casc-conference.org/2019](http://www.casc-conference.org/2019)

This year the 21st International Workshop on Computer Algebra in Scientific Computing (CASC) was held in Moscow (Russia). It took place at Plekhanov Russian University of Economics in the center of Moscow, and the local organizing committee headed by Timur M. Sadykov did a superb job in doing all of the local arrangements.

For the first time not only papers published in the conference proceedings were presented (28 talks), but also ones based on submissions to the journal “Mathematics in Computer Science” (7 talks), and 10 talks were based on reviewed extended abstracts. As in the previous years the presentations were in a single track, all being given in the beautiful “building 1” of Plekhanov Russian University of Economics.



*Conference location*

The tradition of having invited talks was upheld by having a 45 minute presentation of Chee Yap “Towards soft exact computation” on Tuesday morning and Stanislav Poslavsky presenting “An efficient JVM library for commutative algebra” on Thursday morning.

In the business meeting on Tuesday evening the location of next CASC, Johannes-Kepler Universität Linz, Austria, was presented. A shift back to the traditional September date of CASC was suggested.

The excursion on Wednesday afternoon led the participants to the Kremlin.



*Participants of CASC at the excursion to the Kremlin*

The program of the CASC 2019 as well as a link to the online version of the Proceedings (LNCS 11661) may be found on the web site <http://www.casc-conference.org/2019/>.

Andreas Weber (Bonn)

---

## Hinweise auf Konferenzen

---

### Maple Conference

Waterloo, Ontario, Kanada, 15.10. – 17.10.2019

[www.maplesoft.com/mapleconference/](http://www.maplesoft.com/mapleconference/)

This conference is dedicated to exploring different aspects of the math software Maple, including Maple's impact on education, new symbolic computation algorithms and techniques, and the wide range of Maple applications.

Program:

- A keynote presentation by theoretical physicist, Dr. Marvin Weinsten, at the renowned Perimeter Institute.
- A keynote presentation by Maplesoft CEO and President, Dr. Laurent Bernardin, at the University of Waterloo's Federation Hall.
- As part of the scientific program, watch presentations by colleagues and peers. The scientific program is peer reviewed and will be published by Springer.
- Attend workshops and presentations by Maple developers on topics that include: Tensor Calculus, Computer Algebra for Theoretical Physics, Multivariate Limit Computations, Tools for Practice and Assessment, and much more.
- Learn from experienced users, inspire others, and network with colleagues and peers in a social setting.
- Meet Maple Developers face to face and learn about what's coming in Maple.

### MACIS 2019

Gebze-Istanbul, Türkei, 13.11. – 15.11.2019

[macis2019.gtu.edu.tr](http://macis2019.gtu.edu.tr)

MACIS (Mathematical Aspects of Computer and Information Sciences) is a series of biennial conferences focusing on research in mathematical and computational aspects of computing and information science. It is broadly concerned with algorithms, their complexity and their embedding in larger logical systems. At the algorithmic level, there is the rich interplay along the Numerical/Algebraic/Geometric/Topological axes. At the logical level, there are issues of data organization, interpretation and associated tools. These issues often arise in scientific and engineering computation where we need experimental and case studies to validate or enrich the theory. MACIS is interested in outstanding and emerging problems in all these areas.

The 8th iteration of MACIS will take place at Gebze Technical University in Gebze/Istanbul (Turkey) in 13-15 November 2019. Previously MACIS was held in Vienna (2017), Berlin (2015), Nanning (2013), Beijing (2011), Fukuoka (2009), Paris (2007) and Beijing 2006.

### CoCoA 2020 - International School and Workshop on Computer Algebra

Universität Hue (Vietnam), 09.03. – 15.03.2020

[cocoa.dhsphue.edu.vn](http://cocoa.dhsphue.edu.vn)

Die CoCoA-Schule richtet sich an Master-Studenten, Doktoranden und Postdoktoranden aus der ganzen Welt, die an Themen aus der kommutativen Algebra oder algebraischen Geometrie arbeiten und das Computeralgebrasystem CoCoA einsetzen wollen. Es wird zwei Kurse mit zugehörigen Tutorien geben:

- (1) Martin Kreuzer und Lorenzo Robbiano, *Border Bases* (Tutorien: John Abbott und Le Ngoc Long)
- (2) Jürgen Herzog und Takayuki Hibi, *Binomial Ideals* (Tutorien: Anna Bigatti und Tran N.K. Linh)

Die CoCoA Schule findet bereits zum elften Mal statt, jedoch erstmals in Vietnam. Neben den Kursen und Tutorien wird auch eine Poster-Session angeboten, in der die Teilnehmer ihre eigenen Arbeiten präsentieren können. Ferner wird die Schule durch einen anschließenden CoCoA-Workshop (13.-15.3.2020) ergänzt, zu dem sich neben 9 eingeladenen Vortragenden bereits eine Reihe namhafter Mathematiker angesagt hat.

Master-Studenten und Doktoranden, die an der CoCoA-Schule teilnehmen möchten, können sich auf der angegebenen Webseite bis zum 31.12.2019 online bewerben. Für eine begrenzte Zahl an Teilnehmern können die lokalen Kosten (Übernachtungen, Mittagessen) übernommen werden.

### GAMM-Jahrestagung 2020

Kassel, 16.03. – 20.03.2020

<https://jahrestagung.gamm-ev.de>

Erstmals seit der gemeinsamen Jahrestagung von GAMM und DMV im Jahr 2016 wird es bei der 91. Jahrestagung der GAMM wieder ein Minisymposium zur Computeralgebra geben. Der Titel lautet "Symbolic computation methods for differential equations, dynamical systems, and control theory - with special emphasis on biochemical problems." Das Minisymposium wird von Werner Seiler (Kassel) und Eva Zerz (Aachen) organisiert und soll am Dienstag, dem 17.03.2020, stattfinden.

### ICMS 2020

Braunschweig, 13.07. – 16.07.2020

[www.icms-conference.org/2020](http://www.icms-conference.org/2020)

The "International Congress of Mathematical Software" (ICMS) is a community of researchers and practitioners centered around "mathematical software" as a scientific activity.

### **ACA 2020**

Athen, Griechenland, 14.07. – 18.07.2020

[math.unm.edu/aca.html](http://math.unm.edu/aca.html)

The 26th Conference on Applications of Computer Algebra (ACA) will be held in Athens, Greece. This event will take place from July 14 to July 18, 2020.

The ACA conference series is devoted to promoting all kinds of computer algebra applications, and encouraging the interaction of developers of computer algebra systems and packages with researchers and users (including scientists, engineers, educators, and mathematicians).

Topics include, but are not limited to, computer algebra in the sciences, engineering, communication, medicine, pure and applied mathematics, education, business and computer science.

### **ISSAC 2020**

Kalamata, Griechenland, 20.07. – 23.07.2020

[www.issac-conference.org/2020](http://www.issac-conference.org/2020)

The International Symposium on Symbolic and Algebraic Computation (ISSAC) is the premier conference for research in symbolic computation and computer algebra. ISSAC 2019 will be the 44th meeting in the series, which started in 1966 and has been held annually since 1981. The conference presents a range of invited speakers, tutorials, poster sessions, software demonstrations and vendor exhibits with a center-piece of contributed research papers.

ISSAC 2020 will be held on 20-23 July 2020, at Kalamata, Messinia, Greece.

### **CASC 2020**

Linz, Österreich, 14.09. – 18.09.2020

[www.casc-conference.org](http://www.casc-conference.org)

The 22nd International Workshop on Computer Algebra in Scientific Computing, CASC 2020, will be held in Linz, Austria, September 14 - 18, 2020.

The tools of Scientific Computing play an important role in the natural sciences and engineering. Computer Algebra Systems and the underlying algorithms for Symbolic Computation play an increasingly important role within Scientific Computation. The CASC workshop series has been running for over two decades to explore the interaction of these topics, their implementation, and their application.

### **DMV-Jahrestagung 2020**

Chemnitz, 14.09. – 18.09.2020

[www.mathematik.de/dmv/jahrestagungen](http://www.mathematik.de/dmv/jahrestagungen)

Die DMV-Jahrestagung 2020 wird vom 14.-18.9. in Chemnitz stattfinden. Weitere Informationen folgen.

### **GI-Jahrestagung 2020**

Karlsruhe

[www.informatik2020.de](http://www.informatik2020.de)

Die Jahrestagung INFORMATIK 2020 wird in Karlsruhe stattfinden. Das Datum lag zu Redaktionsschluss des Rundbriefs noch nicht vor.

# Antrag auf Mitgliedschaft in der Fachgruppe Computeralgebra der GI in Kooperation mit der DMV und GAMM und auf Bezug des Computeralgebra-Rundbriefs

Bitte zurücksenden an:

Prof. Dr. Wolfram Koepf  
Universität Kassel  
FB Mathematik/Informatik  
Heinrich-Plett-Str. 40  
D-34132 Kassel



Name:	Vorname:
Akadem. Grad:	Geburtsjahr:
<i>Privatanschrift:</i>	
Straße/Postfach:	PLZ Ort:
Telefon:	Telefax:
<i>Dienstanschrift:</i>	
Firma/Institut:	Abteilung:
Straße/Postfach:	PLZ Ort:
Telefon:	Telefax:
E-Mail:	
Gewünschte Postanschrift: <input type="checkbox"/> Privatanschrift <input type="checkbox"/> Dienstanschrift	
Gewünschte Regionalgruppenzuordnung: (http://regionalgruppen.gi.de)	

- ☐ Ich bin persönliches Mitglied der GI und beantrage die Mitgliedschaft in der Fachgruppe Computeralgebra sowie den Bezug des Rundbriefs
- ☐ Ich beantrage assoziierte Mitgliedschaft in der GI und Mitgliedschaft in der Fachgruppe Computeralgebra sowie den Bezug des Rundbriefs
- ☐ ab 1. Januar .....
- ☐ rückwirkend zum 1. Januar des laufenden Jahres (bis zum 30. September möglich).

Ich ordne mich folgender Jahresbeitragsklasse zu:

- ☐ 7,50 Euro für Mitglieder der ☐ GI ☐ DMV ☐ GAMM,

Mitgliedsnummer: .....

- ☐ 7,50 Euro. Ich beantrage gleichzeitig Mitgliedschaft in der ☐ GI ☐ DMV ☐ GAMM und bitte um Zusendung der dazu erforderlichen Unterlagen.

- ☐ 9,00 Euro für Nichtmitglieder. Ich bitte um Zusendung von Informationen über ☐ GI ☐ DMV ☐ GAMM.

- ☐ Ich bitte lediglich um Aktualisierung meiner Adressdaten sowie meiner Angaben über die Zusendung von Informationen.

Ich nehme zur Kenntnis, dass die Aufnahme in die Fachgruppe Computeralgebra zum 1.1. erfolgt und dass die Mitgliedschaft zum 31.12. mit Frist 30.11. schriftlich gekündigt werden kann.

## Datennutzung

Meine oben angegebenen personenbezogenen Daten werden im Rahmen meiner Mitgliedschaft soweit gesetzlich erlaubt oder aufgrund meiner Einwilligung durch die GI oder durch Dritte nach Weitergabe durch die GI wie folgt genutzt:

- ☐ für alle GI-gesellschaftsinternen Aussendungen,
- ☐ für von der GI ausgewählte Informationen mit Bezug zur Informatik, z.B. Weiterbildungsangebote, Informatikveranstaltungen oder -kongresse mit und ohne GI-Beteiligung sowie Publikationen mit Informatikbezug.

Wenn Sie uns Ihre E-Mail-Adresse angegeben haben, wird die Kommunikation soweit möglich elektronisch ausgeführt.

- ☐ Der Nutzung meiner E-Mail-Adresse zu Zwecken, die über die satzungsgemäßen Ziele der GI hinausgehen (wie z.B. Werbung, Markt- und Meinungsforschung) stimme ich zu.

Natürlich können Sie Ihre Zustimmung jederzeit widerrufen oder Ihre E-Mail-Adresse in unserem System löschen lassen, kurze Nachricht an [mitgliederservice@gi.de](mailto:mitgliederservice@gi.de), per Post oder Fax genügt.

Datum: ..... Unterschrift: .....

Rückfragen: Telefon +49 (0)228-302-151/-149 Telefax +49 (0)228-302-167 E-Mail: [mitgliederservice@gi.de](mailto:mitgliederservice@gi.de) <http://gi.de>

---

## Fachgruppenleitung Computeralgebra 2017–2020

---

**Sprecher:**

Prof. Dr. Gregor Kemper  
Zentrum Mathematik – M11  
Technische Universität München  
Boltzmannstr. 3, 85748 Garching  
089-289-17454, -17457 (Fax)  
[kemper@ma.tum.de](mailto:kemper@ma.tum.de)  
<http://www.groups.ma.tum.de/algebra/kemper>

**Vertreterin der GI:**

Prof. Dr. Erika Abraham  
Fachgruppe Informatik  
RWTH Aachen University  
Ahornstr. 55, 52056 Aachen  
0241-80-21242, -22243 (Fax)  
[abraham@cs.rwth-aachen.de](mailto:abraham@cs.rwth-aachen.de)  
<https://ths.rwth-aachen.de/people/erika-abraham/>

**Fachreferent CA-Systeme und -Bibliotheken:**

Prof. Dr. Claus Fieker  
Fachbereich Mathematik  
Technische Universität Kaiserslautern  
Gottlieb-Daimler-Straße, 67663 Kaiserslautern  
0631-205-2392, -4427 (Fax)  
[fieker@mathematik.uni-kl.de](mailto:fieker@mathematik.uni-kl.de)  
<http://www.mathematik.uni-kl.de/~fieker>

**Fachreferent Physik:**

Dr. Thomas Hahn  
Max-Planck-Institut für Physik  
Föhringer Ring 6, 80805 München  
089-32354-300, -304 (Fax)  
[hahn@feynarts.de](mailto:hahn@feynarts.de)  
<http://wwwth.mpp.mpg.de/members/hahn>

**Fachreferent CA an der Hochschule:**

Prof. Dr. Jürgen Klüners  
Mathematisches Institut der Universität Paderborn  
Warburger Str. 100, 33098 Paderborn  
05251-60-2646, -3516 (Fax)  
[klueners@math.uni-paderborn.de](mailto:klueners@math.uni-paderborn.de)  
<https://math.uni-paderborn.de/ag/klueners/>

**Fachreferent Themen, Anwendungen und Publikationen:**

Prof. Dr. Martin Kreuzer  
Fakultät für Informatik und Mathematik  
Universität Passau  
Innstr. 33, 94030 Passau  
0851-509-3120, -3122 (Fax)  
[martin.kreuzer@uni-passau.de](mailto:martin.kreuzer@uni-passau.de)  
<http://www.fim.uni-passau.de/~kreuzer>

**Fachexperte Redaktion Rundbrief:**

Dr. Fabian Reimers  
Zentrum Mathematik – M11  
Technische Universität München  
Boltzmannstr. 3, 85748 Garching  
089-289-17474  
[reimers@ma.tum.de](mailto:reimers@ma.tum.de)  
<http://www.groups.ma.tum.de/algebra/reimers>

**Stellvertretende Sprecherin:**

Prof. Dr. Anne Frühbis-Krüger  
Institut für Algebraische Geometrie  
Welfengarten 1, 30167 Hannover  
0511-762-3592  
[fruehbis-krueger@math.uni-hannover.de](mailto:fruehbis-krueger@math.uni-hannover.de)  
<http://www.iag.uni-hannover.de/~anne>

**Fachexpertin Industrie:**

Xenia Bogomolec  
Coding Services Hannover  
Engelbosteler Damm 15, 30167 Hannover  
0173-3031816  
[indigomind@protonmail.ch](mailto:indigomind@protonmail.ch)  
<https://quant-x-sec.com>

**Fachreferent Sonderforschungsbereich 195:**

Prof. Dr. Meinolf Geck  
Universität Stuttgart  
Institut für Algebra und Zahlentheorie  
Pfaffenwaldring 57, 70569 Stuttgart  
0711 685-65367  
[meinolf.geck@mathematik.uni-stuttgart.de](mailto:meinolf.geck@mathematik.uni-stuttgart.de)  
<http://www.mathematik.uni-stuttgart.de/~geckmf/>

**Fachreferent Themen, Anwendungen und Publikationen:**

Prof. Dr. Florian Heß  
Carl-von-Ossietzky Universität Oldenburg  
Institut für Mathematik, 26111 Oldenburg  
0441-798-2906, -3004 (Fax)  
[florian.hess@uni-oldenburg.de](mailto:florian.hess@uni-oldenburg.de)  
<http://www.staff.uni-oldenburg.de/florian.hess>

**Vertreter der DMV:**

Prof. Dr. Wolfram Koepf  
Institut für Mathematik  
Universität Kassel  
Heinrich-Plett-Str. 40, 34132 Kassel  
0561-804-4207, -4646 (Fax)  
[koepf@mathematik.uni-kassel.de](mailto:koepf@mathematik.uni-kassel.de)  
<http://www.mathematik.uni-kassel.de/~koepf>

**Fachreferent Schule und Didaktik:**

StD Jan Hendrik Müller  
Rivius-Gymnasium der Stadt Attendorf  
Westwall 48, 57439 Attendorf  
02722-5953 (Sekretariat)  
[jan.mueller@math.uni-dortmund.de](mailto:jan.mueller@math.uni-dortmund.de)  
[www.mathebeimueller.de](http://www.mathebeimueller.de)

**Vertreterin der GAMM:**

Prof. Dr. Eva Zerz  
Lehrstuhl D für Mathematik  
RWTH Aachen  
Pontdriesch 14/16, 52062 Aachen  
0241-80-94544, -92108 (Fax)  
[eva.zerz@math.rwth-aachen.de](mailto:eva.zerz@math.rwth-aachen.de)  
<http://www.math.rwth-aachen.de/~Eva.Zerz/>





# Sie sind noch kein Maple User?

Lernen Sie hier, was Sie und vor allem Ihre Schüler und Studenten verpassen!

**Leistungsstarke Mathematik Software, die gleichzeitig leicht zu bedienen ist**

## Warum das interessant ist?

Hören Sie Leuten zu, die sich gewünscht hätten, Maple in Ihrer Schul- oder Studienzeit gehabt zu haben, es aber leider nicht hatten. Entdecken Sie die vielen verschiedenen Möglichkeiten, wie Maple Ihnen und Ihren Schülern und Studenten helfen wird.

Laden Sie sich das kostenlose Whitepaper hier runter:

**[www.maplesoft.com/CAR2019](http://www.maplesoft.com/CAR2019)**



**Neues Maple 2019 jetzt verfügbar!**