int³

Hilfe!

Die Kryptokalypse kommt!

Nimmt mir der Quantencomputer mein Homebanking weg?

Hilfe!

Die Kryptokalypse kommt!

Nimmt mir der Quantencomputer mein Homebanking weg?

David Fuhr, CTO, intcube GmbH

me:

0xdhf

Quantengeometer

Gestalttherapeut

Kryptographie

@SecMan

int³

www.intcube.io

Quantum Algorithm Zoo

This is a comprehensive catalog of quantum algorithms. If you notice any errors or omissions, please email me at stephen.jordan@microsoft.com. (Alternatively, you may submit a pull request to the repository on github.) Your help is appreciated and will be acknowledged.

Algebraic and Number Theoretic Algorithms

Algorithm: Factoring
Speedup: Superpolynomial
Description: Given an n-bit integer, find the prime factorization. The quantum algorithm of Peter Shor solves this in $\widetilde{O}(n^3)$ time [82,125]. The fastest known classical algorithm for integer factorization is the general number field sieve, which is believed to run in time $2^{\widetilde{O}(n^{1/3})}$. The best rigorously proven upper bound on the classical complexity of factoring is $O(2^{n/4+o(1)})$ via the Pollard-Strassen algorithm [252, 362]. Shor's factoring algorithm breaks RSA public-key encryption and the closely related quantum algorithms for discrete logarithms break the DSA and ECDSA digital signature schemes and the Diffie-Hellman key-exchange protocol. A quantum algorithm even faster than Shor's for the special case of factoring "semiprimes", which are widely used in cryptography, is given in [271]. If small factors exist, Shor's algorithm can be beaten by a quantum algorithm using Grover search to speed up the elliptic curve factorization method [366]. Additional optimized versions of Shor's algorithm are given in [384, 386, 431]. There are proposed classical public-key cryptosystems not believed to be broken by quantum algorithms, *cf.* [248]. At the core of Shor's factoring algorithm is order finding, which can be reduced to the Abelian hidden subgroup problem, which is solved using the quantum Fourier transform. A number of other problems are known to reduce to integer factorization including the membership problem for matrix groups over fields of odd order [253], and certain diophantine problems relevant to the synthesis of quantum circuits [254].

Algorithm: Discrete-log
Speedup: Superpolynomial
Description: We are given three n-bit numbers a, b, and N, with the promise that $b = a^s \mod N$ for some s. The task is to find s. As shown by Shor [82], this can be achieved on a quantum computer in poly(n) time. The fastest known classical algorithm requires time superpolynomial in n. By similar techniques to those in [82], quantum computers can solve the discrete logarithm problem on elliptic curves, thereby breaking elliptic curve cryptography [109, 14]. Further optimizations to Shor's algorithm are given in [385, 432]. The superpolynomial quantum speedup has also been extended to the discrete logarithm problem on semigroups [203, 204]. See also Abelian hidden subgroup.

Algorithm: Pell's Equation
Speedup: Superpolynomial
Description: Given a positive nonsquare integer d, Pell's equation is $x^2 - dy^2 = 1$. For any such d there are infinitely many pairs of integers (x,y) solving this equation. Let (x_1, y_1) be the pair that minimizes $x + y\sqrt{d}$. If d is an n-bit integer (*i.e.* $0 \leq d < 2^n$), (x_1, y_1) may in general require exponentially many bits to write down. Thus it is in general impossible to find (x_1, y_1) in polynomial time. Let $R = \log(x_1 + y_1\sqrt{d})$. $\lfloor R \rceil$ uniquely identifies (x_1, y_1). As shown by Hallgren [49], given a n-bit number d, a quantum computer can find $\lfloor R \rceil$ in poly(n) time. No polynomial time classical algorithm for this problem is known. Factoring reduces to this problem. This algorithm breaks the Buchman-Williams cryptosystem. See also Abelian hidden subgroup.

Algorithm: Principal Ideal
Speedup: Superpolynomial
Description: We are given an n-bit integer d and an invertible ideal I of the ring $\mathbb{Z}[\sqrt{d}]$. I is a principal ideal if there exists $\alpha \in \mathbb{Q}(\sqrt{d})$ such that $I = \alpha \mathbb{Z}[\sqrt{d}]$. α may be exponentially large in d. Therefore α cannot in general even be written down in polynomial time. However, $\lfloor \log \alpha \rceil$ uniquely identifies α. The task is to determine whether I is principal and if so find $\lfloor \log \alpha \rceil$. As shown by Hallgren, this can be done in polynomial time on a quantum computer [49]. A modified quantum algorithm for this problem using fewer qubits was given in [131]. A quantum algorithm solving the principal ideal problem in number fields of arbitrary degree (*i.e.* scaling polynomially in the degree) was subsequently given in [329]. Factoring reduces to solving Pell's equation, which reduces to the principal ideal problem. Thus the principal ideal problem is at least as hard as factoring and therefore is probably not in P. See also Abelian hidden subgroup.

Algorithm: Unit Group
Speedup: Superpolynomial
Description: The number field $\mathbb{Q}(\theta)$ is said to be of degree d if the lowest degree polynomial of which θ is a root has degree d. The set $\mathcal{O}$ of elements of $\mathbb{Q}(\theta)$ which are roots of monic polynomials in $\mathbb{Z}[x]$ forms a ring, called the ring of integers of $\mathbb{Q}(\theta)$. The set of units (invertible elements) of the ring $\mathcal{O}$ form a group denoted $\mathcal{O}^*$. As shown by Hallgren [50], and independently by Schmidt and Vollmer [116], for any $\mathbb{Q}(\theta)$ of fixed degree, a quantum computer can find in polynomial time a set of generators for $\mathcal{O}^*$ given a description of θ. No polynomial time classical algorithm for this problem is known. Hallgren and collaborators subsequently discovered how to achieve polynomial scaling in the degree [213]. See also [329]. The algorithms rely on solving Abelian hidden subgroup problems over the additive group of real numbers.

Algorithm: Class Group
Speedup: Superpolynomial
Description: The number field $\mathbb{Q}(\theta)$ is said to be of degree d if the lowest degree polynomial of which θ is a root has degree d. The set $\mathcal{O}$ of elements of $\mathbb{Q}(\theta)$ which are roots of monic polynomials in $\mathbb{Z}[x]$ forms a ring, called the ring of integers of $\mathbb{Q}(\theta)$, which is a Dedekind domain. For a Dedekind domain, the nonzero fractional ideals modulo the nonzero principal ideals form a group called the class group. As shown by Hallgren [50], a quantum computer can find a set of generators for the class group of the ring of integers of any constant degree number field, given a description of θ, in time poly(log($|\mathcal{O}|$)). An improved quantum algorithm, whose runtime is also polynomial in d was subsequently given in [329]. No polynomial time classical algorithm for these problems are known. See also Abelian hidden subgroup.

Algorithm: Gauss Sums
Speedup: Superpolynomial
Description: Let $\mathbb{F}_q$ be a finite field. The elements other than zero of $\mathbb{F}_q$ form a group $\mathbb{F}_q^\times$ under multiplication, and the elements of $\mathbb{F}_q$ form an (Abelian but not necessarily cyclic) group $\mathbb{F}_q^+$ under addition. We can choose some character $\chi^\times$ of $\mathbb{F}_q^\times$ and some character χ^+ of $\mathbb{F}_q^+$. The corresponding Gauss sum is the inner product of these characters: $\sum_{x \neq 0 \in \mathbb{F}_q} \chi^+(x)\chi^\times(x)$ As shown by van Dam and Seroussi [90], Gauss sums can be estimated to polynomial precision on a quantum computer in polynomial time. Although a finite ring does not form a group under multiplication, its set of units does. Choosing a representation for the additive group of the ring, and choosing a representation for the multiplicative group of its units, one can obtain a Gauss sum over the units of a finite ring. These can also be estimated to polynomial precision on a quantum computer in polynomial time [90]. No polynomial time classical algorithm for estimating Gauss sums is known. Discrete log reduces to Gauss sum estimation [90]. Certain partition functions of the Potts model can be computed by a polynomial-time quantum algorithm related to Gauss sum estimation [47].

Algorithm: Primality Proving
Speedup: Polynomial
Description: Given an n-bit number, return a proof of its primality. The fastest classical algorithms are AKS, the best versions of which [393, 394] have essentially-quartic complexity, and ECPP, where the heuristic complexity of the fastest version [395] is also essentially quartic. The fastest known quantum algorithm for this problem is the method of Donis-Vela and Garcia-Escartin [396], with complexity $O(n^2(\log n)^3 \log \log n)$. This improves upon a prior factoring-based quantum algorithm for primality proving [397] that has complexity $O(n^3 \log n \log \log n)$. A recent result of Harvey and Van Der Hoeven [398] can be used to improve the complexity of the factoring-based quantum algorithm for primality proving to $O(n^3 \log n)$ and it may be possible to similarly reduce the complexity of the Donis-Vela-Garcia-Escartin algorithm to $O(n^2(\log n)^3)$ [399].

Algorithm: Solving Exponential Congruences
Speedup: Polynomial
Description: We are given $a, b, c, f, g \in \mathbb{F}_q$. We must find integers x, y such that $af^x + bg^y = c$. As shown in [111], quantum computers can solve this problem in $\widetilde{O}(q^{3/8})$ time whereas the best classical algorithm requires $\widetilde{O}(q^{9/8})$ time. The quantum algorithm of [111] is based on the quantum algorithms for discrete logarithms and searching.

Algorithm: Matrix Elements of Group Representations
Speedup: Superpolynomial
Description: All representations of finite groups and compact linear groups can be expressed as unitary matrices given an appropriate choice of basis. Conjugating the regular representation of a group by the quantum Fourier transform circuit over that group yields a direct sum of the group's irreducible representations. Thus, the efficient quantum Fourier transform over the symmetric group [196], together with the Hadamard test, yields a fast quantum algorithm for additively approximating individual matrix elements of the arbitrary irreducible representations of S_n. Similarly, using the quantum Schur transform [197], one can efficiently approximate matrix elements of the irreducible representations of SU(n) that have polynomial weight. Direct implementations of individual irreducible representations for the groups U(n), SU(n), SO(n), and A_n by efficient quantum circuits are given in [106]. Instances that appear to be exponentially hard for known classical algorithms are also identified in [106].

Algorithm: Verifying Matrix Products
Speedup: Polynomial
Description: Given three $n \times n$ matrices, A,B, and C, the matrix product verification problem is to decide whether AB=C. Classically, the best known algorithm achieves this in time $O(n^2)$, whereas the best known classical algorithm for matrix multiplication runs in time $O(n^{2.373})$. Ambainis *et al.* discovered a quantum algorithm for this problem with runtime $O(n^{7/4})$ [6]. Subsequently, Buhrman and Špalek improved upon this, obtaining a quantum algorithm for this problem with runtime $O(n^{5/3})$ [19]. This latter algorithm is based on results regarding quantum walks that were proven in [85].

Algorithm: Subset-sum
Speedup: Polynomial
Description: Given a list of integers $x_1, \ldots, x_n$, and a target integer s, the subset-sum problem is to determine whether the sum of any subset of the given integers adds up to s. This problem is NP-complete, and therefore is unlikely to be solvable by classical or quantum algorithms with polynomial worst-case complexity. In the hard instances the given integers are of order 2^n and much research on subset sum focuses on average case instances in this regime. In [178], a quantum algorithm is given that solves such instances in time $2^{0.241n}$, up to polynomial factors. This quantum algorithm works by applying a variant of Ambainis's quantum walk algorithm for element-distinctness [7] to speed up a sophisticated classical algorithm for this problem due to Howgrave-Graham and Joux. The fastest known classical algorithm for such instances of subset-sum runs in time $2^{0.291n}$, up to polynomial factors [404].

Algorithm: Decoding
Speedup: Varies
Description: Classical error correcting codes allow the detection and correction of bit-flips by storing data reduntantly. Maximum-likelihood decoding for arbitrary linear codes is NP-complete in the worst case, but for structured codes or bounded error efficient decoding algorithms are known. Quantum algorithms have been formulated to speed up the decoding of convolutional codes [238] and simplex codes [239].

Algorithm: Quantum Cryptanalysis

Navigation

Algebraic & Number Theoretic
Oracular
Approximation and Simulation
Optimization, Numerics, & Machine Learning
Acknowledgments
References

Translations

This page has been translated into:
Japanese
Chinese

Other Surveys

For overviews of quantum algorithms I recommend:

Nielsen and Chuang
Childs
Preskill
Mosca
Childs and van Dam
van Dam and Sasaki
Bacon and van Dam
Montanaro
Hidary

Terminology

If there exists a positive constant α such that the runtime $C(n)$ of the best known classical algorithm and the runtime $Q(n)$ of the quantum algorithm satisfy $C = 2^{\Omega(Q^\alpha)}$ then I call the speedup superpolynomial. Otherwise I call it polynomial. For a review of the $O, \Omega, \Theta, \widetilde{O}, \ldots$ notations see the Wikipedia article.

About

Author:

Stephen Jordan
Microsoft Quantum

Last updated: October 14, 2022
Date created: April 22, 2011

Oracular Algorithms

Approximation and Simulation Algorithms

Optimization, Numerics, and Machine Learning

Quantum Algorithm Zoo

This is a comprehensive catalog of quantum algorithms. If you notice any errors or omissions, please email me at stephen.jordan@microsoft.com. (Alternatively, you may submit a pull request to the repository on github.) Your help is appreciated and will be acknowledged.

Algebraic and Number Theoretic Algorithms

Algorithm: Factoring
Speedup: Superpolynomial
Description: Given an n-bit integer, find the prime factorization. The quantum algorithm of Peter Shor solves this in $\widetilde{O}(n^3)$ time [82,125]. The fastest known classical algorithm for integer factorization is the general number field sieve, which is believed to run in time $2^{\widetilde{O}(n^{1/3})}$. The best rigorously proven upper bound on the classical complexity of factoring is $O(2^{n^{0.4+o(1)}})$ via the Pollard-Strassen algorithm [252, 362]. Shor's factoring algorithm breaks RSA public-key encryption and the closely related quantum algorithms for discrete logarithms break the DSA and ECDSA digital signature schemes and the Diffie-Hellman key-exchange protocol. A quantum algorithm even faster than Shor's for the special case of factoring "semiprimes", which are widely used in cryptography, is given in [271]. If small factors exist, Shor's algorithm can be beaten by a quantum algorithm using Grover search to speed up the elliptic curve factorization method [366]. Additional optimized versions of Shor's algorithm are given in [384, 386, 431]. There are proposed classical public-key cryptosystems not believed to be broken by quantum algorithms, *cf.* [248]. At the core of Shor's factoring algorithm is order finding, which can be reduced to the Abelian hidden subgroup problem, which is solved using the quantum Fourier transform. A number of other problems are known to reduce to integer factorization including the membership problem for matrix groups over fields of odd order [253], and certain diophantine problems relevant to the synthesis of quantum circuits [254].

Algorithm: Discrete-log
Speedup: Superpolynomial
Description: We are given three n-bit numbers a, b, and N, with the promise that $b = a^s \mod N$ for some s. The task is to find s. As shown by Shor [82], this can be achieved on a quantum computer in poly(n) time. The fastest known classical algorithm requires time superpolynomial in n. By similar techniques to those in [82], quantum computers can solve the discrete logarithm problem on elliptic curves, thereby breaking elliptic curve cryptography [109, 14]. Further optimizations to Shor's algorithm are given in [385, 432]. The superpolynomial quantum speedup has also been extended to the discrete logarithm problem on semigroups [203, 204]. See also Abelian hidden subgroup.

Algorithm: Pell's Equation
Speedup: Superpolynomial
Description: Given a positive nonsquare integer d, Pell's equation is $x^2 - dy^2 = 1$. For any such d there are infinitely many pairs of integers (x,y) solving this equation. Let (x_1, y_1) be the pair that minimizes $x + y\sqrt{d}$. If d is an n-bit integer (*i.e.* $0 \leq d < 2^n$), (x_1, y_1) may in general require exponentially many bits to write down. Thus it is in general impossible to find (x_1, y_1) in polynomial time. Let $R = \log(x_1 + y_1\sqrt{d})$. $\lfloor R \rceil$ uniquely identifies (x_1, y_1). As shown by Hallgren [49], given a n-bit number d, a quantum computer can find $\lfloor R \rceil$ in poly(n) time. No polynomial time classical algorithm for this problem is known. Factoring reduces to this problem. This algorithm breaks the Buchman-Williams cryptosystem. See also Abelian hidden subgroup.

Algorithm: Principal Ideal
Speedup: Superpolynomial
Description: We are given an n-bit integer d and an invertible ideal I of the ring $\mathbb{Z}[\sqrt{d}]$. I is a principal ideal if there exists $\alpha \in \mathbb{Q}(\sqrt{d})$ such that $I = \alpha\mathbb{Z}[\sqrt{d}]$. α may be exponentially large in d. Therefore α cannot in general even be written down in polynomial time. However, $\lfloor \log \alpha \rceil$ uniquely identifies α. The task is to determine whether I is principal and if so find $\lfloor \log \alpha \rceil$. As shown by Hallgren, this can be done in polynomial time on a quantum computer [49]. A modified quantum algorithm for this problem using fewer qubits was given in [131]. A quantum algorithm solving the principal ideal problem in number fields of arbitrary degree (*i.e.* scaling polynomially in the degree) was subsequently given in [329]. Factoring reduces to solving Pell's equation, which reduces to the principal ideal problem. Thus the principal ideal problem is at least as hard as factoring and therefore is probably not in P. See also Abelian hidden subgroup.

Algorithm: Unit Group
Speedup: Superpolynomial
Description: The number field $\mathbb{Q}(\theta)$ is said to be of degree d if the lowest degree polynomial of which θ is a root has degree d. The set $\mathcal{O}$ of elements of $\mathbb{Q}(\theta)$ which are roots of monic polynomials in $\mathbb{Z}[x]$ forms a ring, called the ring of integers of $\mathbb{Q}(\theta)$. The set of units (invertible elements) of the ring $\mathcal{O}$ form a group denoted $\mathcal{O}^*$. As shown by Hallgren [50], and independently by Schmidt and Vollmer [116], for any $\mathbb{Q}(\theta)$ of fixed degree, a quantum computer can find in polynomial time a set of generators for $\mathcal{O}^*$ given a description of θ. No polynomial time classical algorithm for this problem is known. Hallgren and collaborators subsequently discovered how to achieve polynomial scaling in the degree [213]. See also [329]. The algorithms rely on solving Abelian hidden subgroup problems over the additive group of real numbers.

Algorithm: Class Group
Speedup: Superpolynomial
Description: The number field $\mathbb{Q}(\theta)$ is said to be of degree d if the lowest degree polynomial of which θ is a root has degree d. The set $\mathcal{O}$ of elements of $\mathbb{Q}(\theta)$ which are roots of monic polynomials in $\mathbb{Z}[x]$

Navigation

Algebraic & Number Theoretic
Oracular
Approximation and Simulation
Optimization, Numerics, & Machine Learning
Acknowledgments
References

Translations

This page has been translated into:
Japanese
Chinese

Other Surveys

For overviews of quantum algorithms I recommend:

Nielsen and Chuang
Childs
Preskill
Mosca
Childs and van Dam
van Dam and Sasaki
Bacon and van Dam
Montanaro
Hidary

Terminology

If there exists a positive constant α such that the runtime $C(n)$ of the best known classical algorithm and the runtime $Q(n)$ of the quantum algorithm satisfy $C = 2^{\Omega(Q^\alpha)}$ then I call the speedup superpolynomial. Otherwise I call it polynomial. For a review of the $O, \Omega, \Theta, \widetilde{O}, \ldots$ notations see the Wikipedia article.

About

Author:

Stephen Jordan
Microsoft Quantum

Last updated: October 14, 2022
Date created: April 22, 2011

Quantum Algorithm Zoo

This is a comprehensive catalog of quantum algorithms. If you notice any errors or omissions, please email me at stephen.jordan@microsoft.com. (Alternatively, you may submit a pull request to the repository on github.) Your help is appreciated and will be acknowledged.

Algebraic and Number Theoretic Algorithms

Algorithm: Factoring
Speedup: Superpolynomial
Description: Given an n-bit integer, find the prime factorization. The quantum algorithm of Peter Shor solves this in $\widetilde{O}(n^3)$ time [82,125]. The fastest known classical algorithm for integer factorization is the general number field sieve, which is believed to run in time $2^{\widetilde{O}(n^{1/3})}$. The best rigorously proven upper bound on the classical complexity of factoring is $O(2^{n/4+o(1)})$ via the Pollard-Strassen algorithm [252, 362]. Shor's factoring algorithm breaks RSA public-key encryption and the closely related quantum algorithms for discrete logarithms break the DSA and ECDSA digital signature schemes and the Diffie-Hellman key-exchange protocol. A quantum algorithm even faster than Shor's for the special case of factoring "semiprimes", which are widely used in cryptography, is given in [271]. If small factors exist, Shor's algorithm can be beaten by a quantum algorithm using Grover search to speed up the elliptic curve factorization method [366]. Additional optimized versions of Shor's algorithm are given in [384, 386, 431]. There are proposed classical public-key cryptosystems not believed to be broken by quantum algorithms, *cf.* [248]. At the core of Shor's factoring algorithm is order finding, which can be reduced to the Abelian hidden subgroup problem, which is solved using the quantum Fourier transform. A number of other problems are known to reduce to integer factorization including the membership problem for matrix groups over fields of odd order [253], and certain diophantine problems relevant to the synthesis of quantum circuits [254].

Algorithm: Discrete-log
Speedup: Superpolynomial
Description: We are given three n-bit numbers a, b, and N, with the promise that $b = a^s \mod N$ for some s. The task is to find s. As shown by Shor [82], this can be achieved on a quantum computer in poly(n) time. The fastest known classical algorithm requires time superpolynomial in n. By similar techniques to those in [82], quantum computers can solve the discrete logarithm problem on elliptic curves, thereby breaking elliptic curve cryptography [109, 14]. Further optimizations to Shor's algorithm are given in [385, 432]. The superpolynomial quantum speedup has also been extended to the discrete logarithm problem on semigroups [203, 204]. See also Abelian hidden subgroup.

Oracular Algorithms

Algorithm: Searching
Speedup: Polynomial
Description: We are given an oracle with N allowed inputs. For one input w ("the winner") the corresponding output is 1, and for all other inputs the corresponding output is 0. The task is to find w. On a classical computer this requires $\Omega(N)$ queries. The quantum algorithm of Lov Grover achieves this using $O(\sqrt{N})$ queries [48], which is optimal [216]. This has algorithm has subsequently been generalized to search in the presence of multiple "winners" [15], evaluate the sum of an arbitrary function [15,16,73], find the global minimum of an arbitrary function [35,75, 255], take advantage of alternative initial states [100] or nonuniform probabilistic priors [123], work with oracles whose runtime varies between inputs [138], approximate definite integrals [77], and converge to a fixed-point [208, 209, 433]. Considerations on optimizing the depth of quantum search circuits are given in [405]. The generalization of Grover's algorithm known as amplitude estimation [17] is now an important primitive in quantum algorithms. Amplitude estimation forms the core of most known quantum algorithms related to collision finding and graph properties. One of the natural applications for Grover search is speeding up the solution to NP-complete problems such as 3-SAT. Doing so is nontrivial, because the best classical algorithm for 3-SAT is not quite a brute force search. Nevertheless, amplitude amplification enables a quadratic quantum speedup over the best classical 3-SAT algorithm, as shown in [133]. Quadratic speedups for other constraint satisfaction problems are obtained in [134]. For further examples of application of Grover search and amplitude amplification see [261, 262]. A problem closely related to, but harder than, Grover search, is spatial search, in which database queries are limited by some graph structure. On sufficiently well-connected graphs, $O(\sqrt{n})$ quantum query complexity is still achievable [274,275,303, 304, 305, 306, 330].


https://www.wallpaperflare.com/

Algorithm: Quantum Cryptanalysis
Speedup: Various
Description: It is well-known that Shor's algorithms for factoring and discrete logarithms [82,125] completely break the RSA and Diffie-Hellman cryptosystems, as well as their elliptic-curve-based variants [109, 14]. (A number of "post-quantum" public-key cryptosystems have been proposed to replace these primitives, which are not known to be broken by quantum attacks.) Beyond Shor's algorithm, there is a growing body of work on quantum algorithms specifically designed to attack cryptosystems. These generally fall into three categories. The first is quantum algorithms providing polynomial or sub-exponential time attacks on cryptosystems under standard assumptions. In particular, the algorithm of Childs, Jao, and Soukharev for finding isogenies of elliptic curves breaks certain elliptic curve based cryptosystems in subexponential time that were not already broken by Shor's algorithm [283]. The second category is quantum algorithms achieving polynomial improvement over known classical cryptanalytic attacks by speeding up parts of these classical algorithms using Grover search, quantum collision finding, etc. Such attacks on private-key [284, 285, 288, 315, 316] and public-key [262, 287] primitives, do not preclude the use of the associated cryptosystems but may influence choice of key size. The third category is attacks that make use of quantum superposition queries to block ciphers. These attacks in many cases completely break the cryptographic primitives [286, 289, 290, 291, 292]. However, in most practical situations such superposition queries are unlikely to be feasible.


Kryptokalypse?
Shake Well
Calypso
TASTE OF THE ISLANDS
OCEAN BLUE LEMONADE
STRAWBERRY LEMONADE
ORIGINAL LEMONADE
SOUTHERN PEACH LEMONADE
ISLAND WAVE LEMONADE
SHAKE, WELL
Real Lemon BITS
16 FL OZ (1 PT) (473 mL)

www.dkb.de | DKB CA 1O1 | DigiCert Trusted Root G4

Inhabername

Unternehmenssitz: Land	DE
Unternehmenssitz: Bundesland/Provinz	Berlin
Unternehmenssitz: Ort	Charlottenburg
Organisationsart	Private Organization
Seriennummer	HRB 34165
Land	DE
Bundesland/Provinz	Berlin
Ort	Berlin
Organisation	Deutsche Kreditbank AG
Allgemeiner Name	www.dkb.de

Ausstellername

Land	DE
Organisation	Deutsche Kreditbank AG
Allgemeiner Name	DKB CA 1O1

Gültigkeit

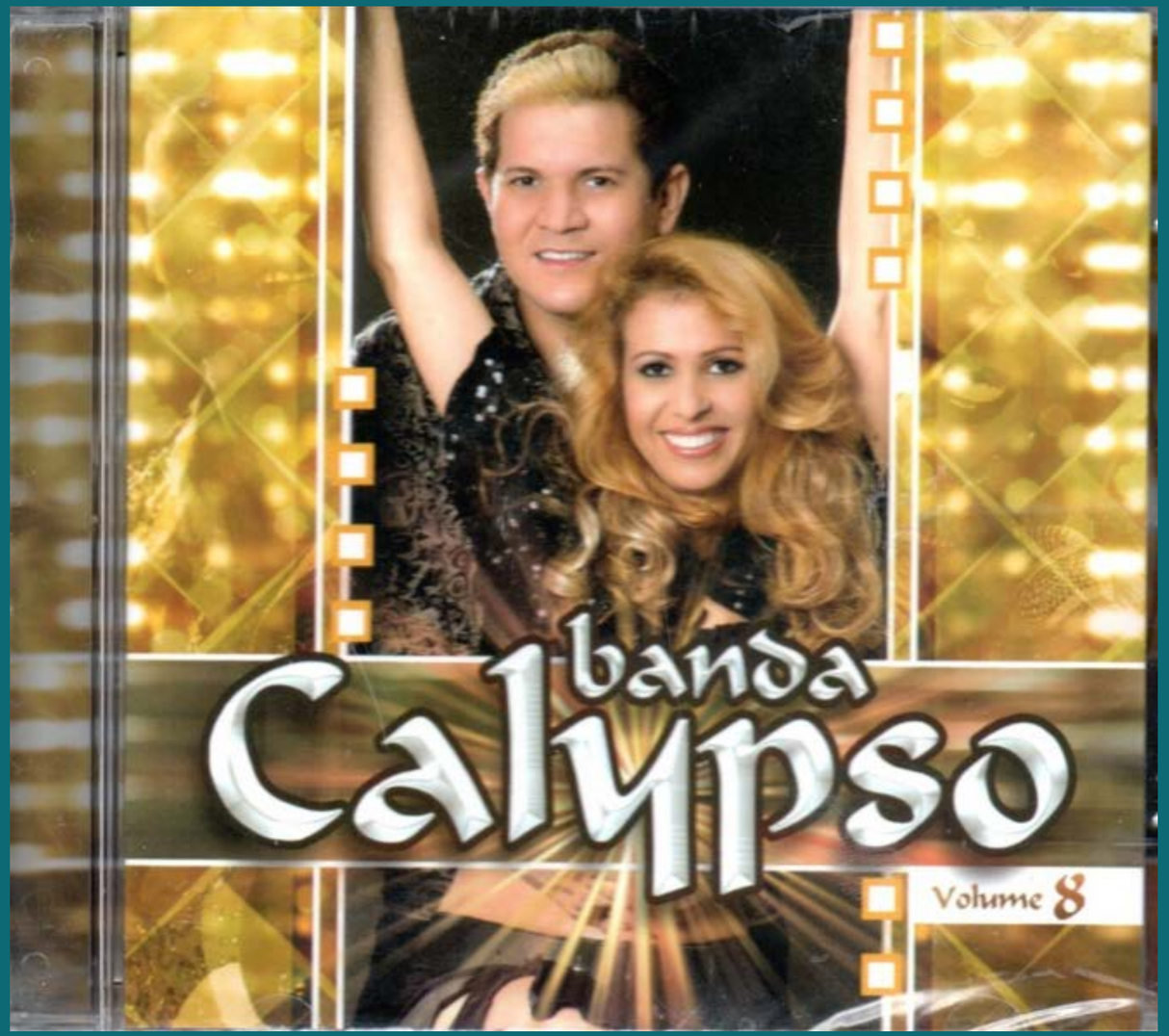
Beginn	Mon, 26 Sep 2022 00:00:00 GMT
Ende	Thu, 26 Oct 2023 23:59:59 GMT

Alternative Inhaberbezeichnungen

DNS-Name	www.dkb.de
DNS-Name	apps.dkb.de

Öffentlicher Schlüssel - Informationen

Algorithmus	RSA
Schlüssellänge	4096


banda
Calypso
Volume 8

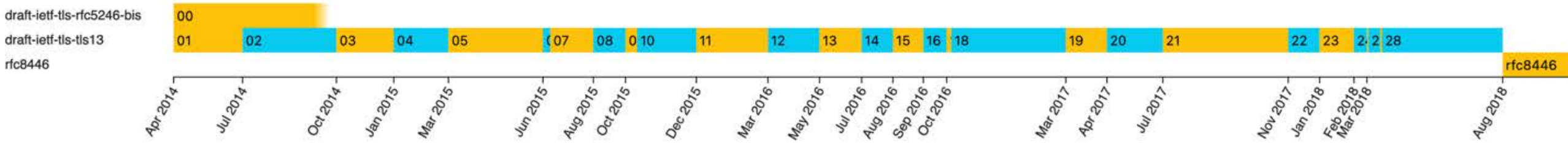
The Transport Layer Security (TLS) Protocol Version 1.3

RFC 8446

Status | IESG evaluation record | IESG writeups | Email expansions | History

Versions:

00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 RFC 8446



Document	**Type**	RFC - Proposed Standard (August 2018) Errata IPR Obsoletes RFC 5077, RFC 5246, RFC 6961 Updates RFC 5705, RFC 6066 Was draft-ietf-tls-tls13 (tls WG)
	Author	Eric Rescorla ✉
	Last updated	2020-03-07
	Replaces	draft-ietf-tls-rfc5246-bis
	RFC stream	Internet Engineering Task Force (IETF)
	Formats	txt html pdf htmlized w/errata bibtex

9.1. Mandatory-to-Implement Cipher Suites

In the absence of an application profile standard specifying otherwise:

A TLS-compliant application MUST implement the TLS_AES_128_GCM_SHA256 [GCM] cipher suite and SHOULD implement the TLS_AES_256_GCM_SHA384 [GCM] and TLS_CHACHA20_POLY1305_SHA256 [RFC8439] cipher suites (see Appendix B.4).

A TLS-compliant application MUST support digital signatures with rsa_pkcs1_sha256 (for certificates), rsa_pss_rsae_sha256 (for CertificateVerify and certificates), and ecdsa_secp256r1_sha256. A TLS-compliant application MUST support key exchange with secp256r1 (NIST P-256) and SHOULD support key exchange with X25519 [RFC7748].

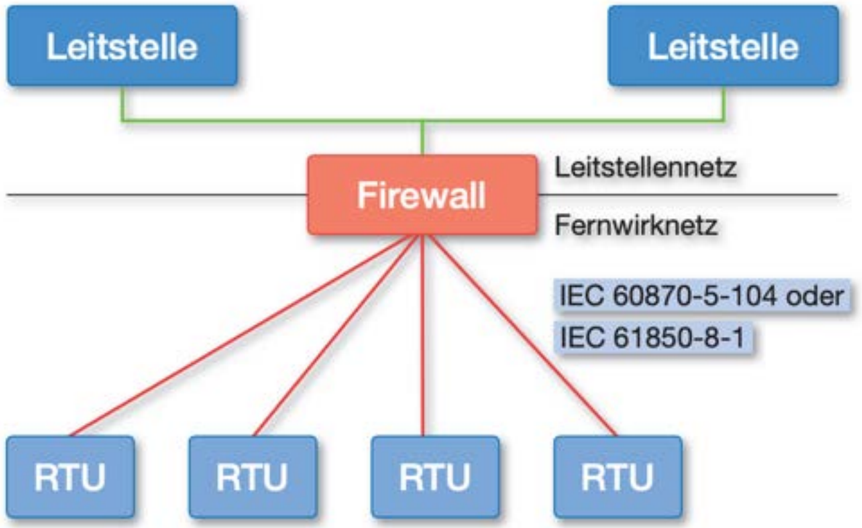

banda
Calypso

```
(base) idh@emmi ~ % ssh -Q cipher
3des-cbc
aes128-cbc
aes192-cbc
aes256-cbc
aes128-ctr
aes192-ctr
aes256-ctr
aes128-gcm@openssh.com
aes256-gcm@openssh.com
chacha20-poly1305@openssh.com
(base) idh@emmi ~ % ssh -Q kex
diffie-hellman-group1-sha1
diffie-hellman-group14-sha1
diffie-hellman-group14-sha256
diffie-hellman-group16-sha512
diffie-hellman-group18-sha512
diffie-hellman-group-exchange-sha1
diffie-hellman-group-exchange-sha256
ecdh-sha2-nistp256
ecdh-sha2-nistp384
ecdh-sha2-nistp521
curve25519-sha256
curve25519-sha256@libssh.org
sntrup761x25519-sha512@openssh.com
(base) idh@emmi ~ % ssh -Q key
ssh-ed25519
ssh-ed25519-cert-v01@openssh.com
ssh-rsa
ssh-dss
ecdsa-sha2-nistp256
ecdsa-sha2-nistp384
ecdsa-sha2-nistp521
ssh-rsa-cert-v01@openssh.com
ssh-dss-cert-v01@openssh.com
ecdsa-sha2-nistp256-cert-v01@openssh.com
ecdsa-sha2-nistp384-cert-v01@openssh.com
ecdsa-sha2-nistp521-cert-v01@openssh.com
(base) idh@emmi ~ %
```


banda
Calypso

Anwendungshinweis für die Normenreihe IEC 62351

Informationssicherheit in der Netz- und Stationsleittechnik

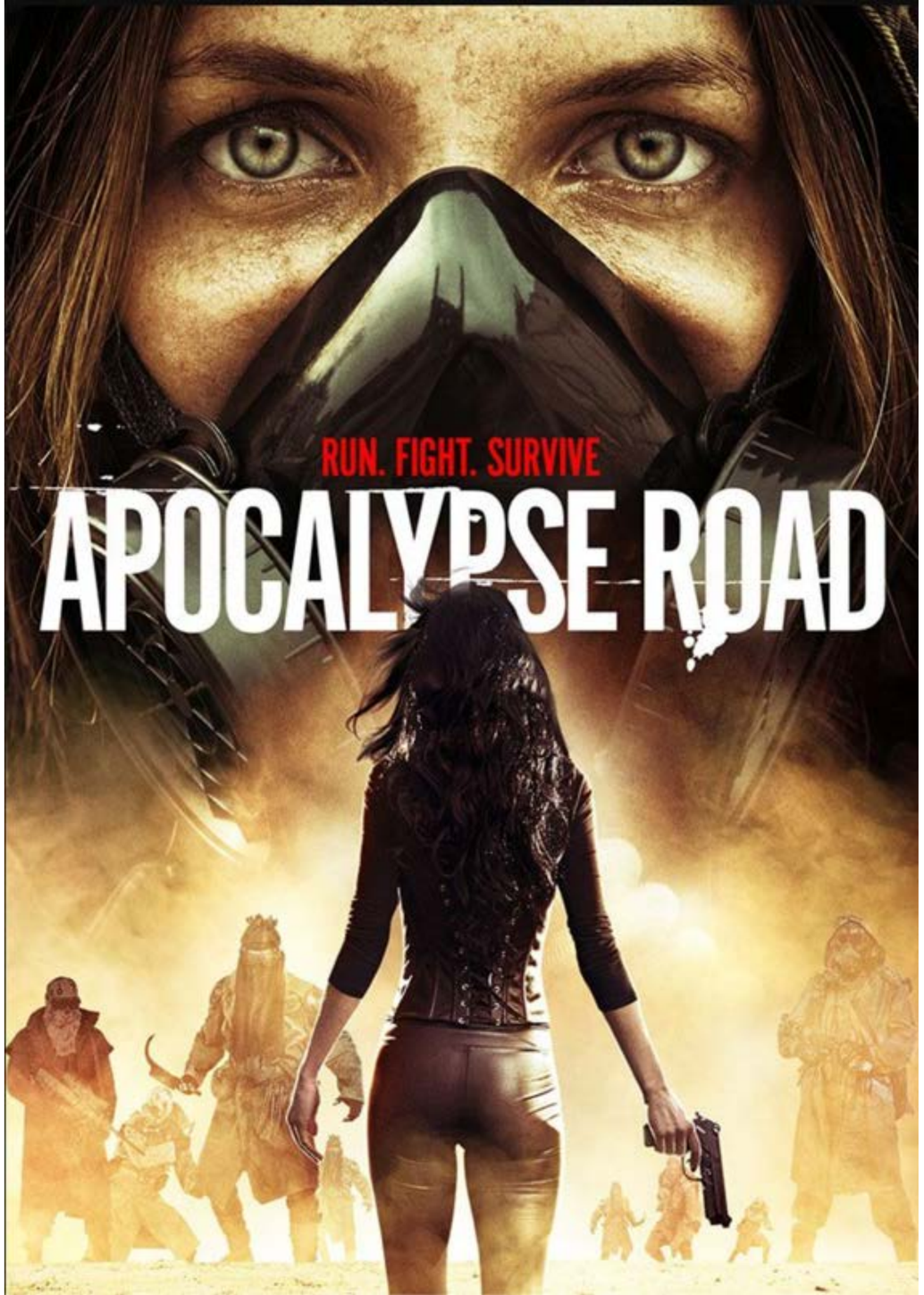


ng 1: Aufbau eines typischen Leit- und Fernwirksystems

Anwendung IEC 62351-3/5

Erfolgt die Kommunikation zwischen der Leitstelle und der Fernwirktechnik mittels IEC 60870-5-104 [2] kann der Normenteil IEC 62351-5 [6] in Verbindung mit Teil IEC 62351-3 [5] zur Absicherung der Kommunikation (Authentizität, Integrität, Vertraulichkeit) eingesetzt werden. Für IP-basierte Protokolle wird durch die IEC 62351 in diesem Fall kein neues Sicherheitsprotokoll definiert, sondern die Anwendung des bekannten SSL/TLS-Protokolls (Transport Layer Security) spezifiziert, das in vielen Anwendungen weit verbreitet ist, wie z. B. Zugriff auf Web Server, Online-Banking, etc. Für Kommunikation nach IEC 60870-5-101 [3] enthält der Teil IEC 62351-5 ein Verfahren zur Integritäts-sicherung der übertragenen Daten.

https://www.dke.de/resource/blob/1806462/2d569bd41f55c4b1726cf33f46be491b/informationssicherheit-in-der-netz--und-stationsleittechnik-download-data.pdf


RUN. FIGHT. SURVIVE
APOCALYPSE ROAD


ethereum-cryptography
npm
npm i ethereum-cryptography
Follow
ethereum-cryptography
npm v1.1.2
Travis CI passing
license MIT
types TypeScript
All pure-js cryptographic primitives normally used when developing Javascript / TypeScript applications and tools for Ethereum.
January 2022 update: We've released v1.0 of the package, a complete rewrite:
6x smaller: ~5,000 lines of code instead of ~24,000 (with all deps); 650KB instead of 10.2MB
5 dependencies by 1 author instead of 38 by 5 authors
Audited by an independent security firm
Check out the article about it: A safer, smaller, and faster Ethereum cryptography stack
Take a glance at the Upgrading section for breaking changes: there are almost none
The cryptographic primitives included are:
Hashes: SHA256, keccak-256, RIPEMD160, BLAKE2b
KDFs: PBKDF2, Scrypt
CSPRNG (Cryptographically strong pseudorandom number generator)
secp256k1 curve
BIP32 HD Keygen
BIP39 Mnemonic phrases
AES Encryption

DEHOGA Ortsverband Braunlage
Ideenwerk 971

Wir fordern:

Die Polizeidienststelle Braunlage muss auch weiterhin rund um die Uhr besetzt bleiben und die Personalstärke darf auf keinen Fall verringert werden!

Es kann nicht hingenommen werden, dass 1 Million Übernachtungsgäste sich nachts selbst überlassen werden sollen. Ebenso sind Handel, Handwerk und Gastronomie und die vielen neuen Investoren betroffen. Nach vielen Jahren der Stagnation geht es in unserer Gemeinde endlich wieder deutlich sichtbar aufwärts. Wollen wir das Alles durch eine nicht vorhandene Polizeipräsenz wieder infrage stellen?

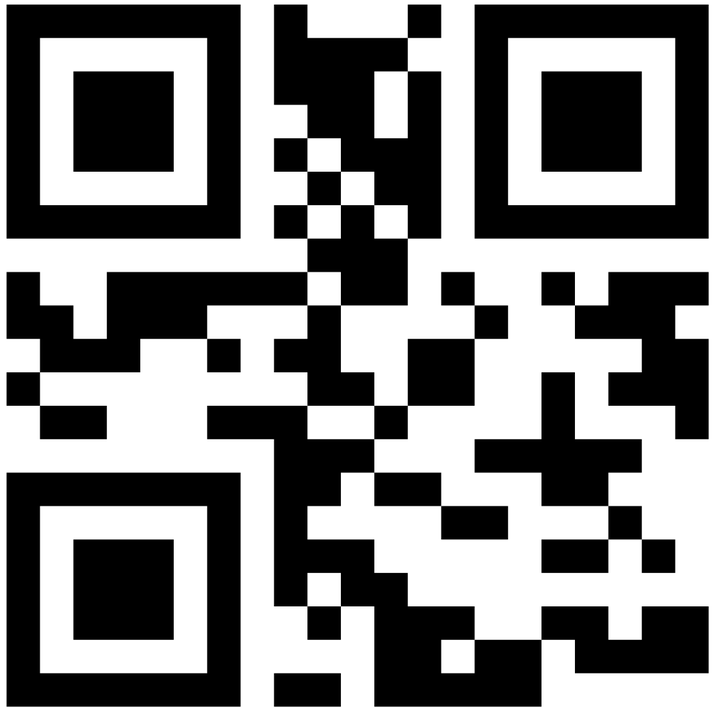
Wir sagen: NEIN!!!

Name:	Vorname:	Adresse:	Unterschrift:

„Unsere Polizei muss bleiben" – eine Initiative von Ideenwerk 971 - Verein zur Wirtschaftsförderung in Braunlage und Umgeburg e.V., Vorsitzender: Burkhard Otto-Küstner, Oberförster-Ulrichs-Straße 16, 38700 Braunlage und DEHOGA Ortsverband Braunlage, Vorsitzende: Jutta Engel, Herzog-Johann-Albrecht-Straße 27, 38700 Braunlage


PRISM


IT HAS
BEGUN!!!!!!!!!
memegenerator.net



David Fuhr, CTO, intcube GmbH
david@intcube.io | twitter.com/0xdhf | linkedin.com/in/davidfuhr