Gerwin Klein

List of Publications by Year
in descending order

74	3,091	471061 17	315357 38
papers	citations	h-index	g-index

77	77	77	1177
all docs	docs citations	times ranked	citing authors

#	Article	IF	Citations
1	Cyberassured Systems Engineering at Scale. IEEE Security and Privacy, 2022, 20, 52-64.	1.5	9
2	Cogent: uniqueness types and certifying compilation. Journal of Functional Programming, 2021, 31, .	0.5	12
3	Formal Reasoning Under Cached Address Translation. Journal of Automated Reasoning, 2020, 64, 911-945.	1.1	9
4	seL4 in Australia. Communications of the ACM, 2020, 63, 72-75.	3.3	14
5	Towards Provable Timing-Channel Prevention. Operating Systems Review (ACM), 2020, 54, 1-7.	1.5	6
6	Can We Prove Time Protection?. , 2019, , .		9
7	A Formal Approach to Constructing Secure Air Vehicle Software. Computer, 2018, 51, 14-23.	1.2	23
8	Formally verified software in the real world. Communications of the ACM, 2018, 61, 68-77.	3.3	30
9	Introduction to Milestones in Interactive Theorem Proving. Journal of Automated Reasoning, 2018, 61, 1-8.	1.1	2
10	Backwards and Forwards with Separation Logic. Lecture Notes in Computer Science, 2018, , 68-87.	1.0	4
11	Bringing Effortless Refinement of Data Layouts to Cogent. Lecture Notes in Computer Science, 2018, , 134-149.	1.0	3
12	Provably trustworthy systems. Philosophical Transactions Series A, Mathematical, Physical, and Engineering Sciences, 2017, 375, 20150404.	1.6	5
13	The Cogent Case for Property-Based Testing. , 2017, , .		5
14	Refinement through restraint: bringing down the cost of verification. , 2016, , .		18
15	A Framework for the Automatic Formal Verification of Refinement from Cogent to C. Lecture Notes in Computer Science, 2016, , 323-340.	1.0	12
16	Interactive Theorem Proving. Journal of Automated Reasoning, 2016, 56, 201-203.	1.1	2
17	Cogent. , 2016, , .		43
18	Cogent. ACM SIGPLAN Notices, 2016, 51, 175-188.	0.2	3

#	Article	IF	Citations
19	CoGENT. Operating Systems Review (ACM), 2016, 50, 175-188.	1.5	6
20	Refinement through restraint: bringing down the cost of verification. ACM SIGPLAN Notices, 2016, 51, 89-102.	0.2	4
21	Cogent. Computer Architecture News, 2016, 44, 175-188.	2.5	4
22	Empirical Study Towards a Leading Indicator for Cost of Formal Software Verification. , 2015, , .		13
23	An empirical research agenda for understanding formal methods productivity. Information and Software Technology, 2015, 60, 102-112.	3.0	15
24	Automated Verification of RPC Stub Code. Lecture Notes in Computer Science, 2015, , 273-290.	1.0	2
25	Don't sweat the small stuff. ACM SIGPLAN Notices, 2014, 49, 429-439.	0.2	10
26	Productivity for proof engineering. , 2014, , .		7
27	Don't sweat the small stuff. , 2014, , .		27
28	Concrete Semantics. , 2014, , .		113
29	Concerned with the unprivileged: user programs in kernel refinement. Formal Aspects of Computing, 2014, 26, 1205-1229.	1.4	7
30	Comprehensive formal verification of an OS microkernel. ACM Transactions on Computer Systems, 2014, 32, 1-70.	0.6	270
31	Proof Engineering Considered Essential. Lecture Notes in Computer Science, 2014, , 16-21.	1.0	6
32	File systems deserve verification too!. Operating Systems Review (ACM), 2014, 48, 58-64.	1.5	1
33	Formal specifications better than function points for code sizing. , 2013, , .		5
34	File systems deserve verification too!. , 2013, , .		16
35	Towards a verified component platform. , 2013, , .		3
36	Translation validation for a verified OS kernel. , 2013, , .		80

#	Article	IF	Citations
37	Formally Verified System Initialisation. Lecture Notes in Computer Science, 2013, , 70-85.	1.0	7
38	Translation validation for a verified OS kernel. ACM SIGPLAN Notices, 2013, 48, 471-482.	0.2	17
39	Large-scale formal verification in practice: A process perspective. , 2012, , .		14
40	Simulation modeling of a large-scale formal verification process. , 2012, , .		8
41	It's Time for Trustworthy Systems. IEEE Security and Privacy, 2012, 10, 67-70.	1.5	13
42	Challenges and Experiences in Managing Large-Scale Proofs. Lecture Notes in Computer Science, 2012, , 32-48.	1.0	17
43	Mechanised Separation Algebra. Lecture Notes in Computer Science, 2012, , 332-337.	1.0	17
44	Bridging the Gap: Automatic Verified Abstraction of C. Lecture Notes in Computer Science, 2012, , 99-115.	1.0	34
45	Noninterference for Operating System Kernels. Lecture Notes in Computer Science, 2012, , 126-142.	1.0	42
46	seL4 Enforces Integrity. Lecture Notes in Computer Science, 2011, , 325-340.	1.0	49
47	seL4. Communications of the ACM, 2010, 53, 107-115.	3.3	254
48	Refinement in the Formal Verification of the seL4 Microkernel. , 2010, , 323-339.		7
49	From a Verified Kernel towards Verified Systems. Lecture Notes in Computer Science, 2010, , 21-33.	1.0	9
50	capDL. , 2010, , .		15
51	The road to trustworthy systems. , 2010, , .		10
52	The L4.verified Project â€" Next Steps. Lecture Notes in Computer Science, 2010, , 86-96.	1.0	6
53	seL4. , 2009, , .		1,046
54	Experience report. , 2009, , .		18

#	Article	IF	Citations
55	Operating System Verification. Journal of Automated Reasoning, 2009, 42, 123-124.	1.1	11
56	Operating system verificationâ€"An overview. Sadhana - Academy Proceedings in Engineering Sciences, 2009, 34, 27-69.	0.8	87
57	Experience report. ACM SIGPLAN Notices, 2009, 44, 91-96.	0.2	5
58	Types, Maps and Separation Logic. Lecture Notes in Computer Science, 2009, , 276-292.	1.0	6
59	Secure Microkernels, State Monads and Scalable Refinement. Lecture Notes in Computer Science, 2008, , 167-182.	1.0	57
60	Verified Protection Model of the seL4 Microkernel. Lecture Notes in Computer Science, 2008, , 99-114.	1.0	23
61	Mapped Separation Logic. Lecture Notes in Computer Science, 2008, , 15-29.	1.0	3
62	Types, bytes, and separation logic. ACM SIGPLAN Notices, 2007, 42, 97-108.	0.2	27
63	Types, bytes, and separation logic. , 2007, , .		92
64	Towards trustworthy computing systems. Operating Systems Review (ACM), 2007, 41, 3-11.	1.5	55
65	On the Automated Synthesis of Proof-Carrying Temporal Reference Monitors. , 2007, , 111-126.		1
66	A machine-checked model for a Java-like language, virtual machine, and compiler. ACM Transactions on Programming Languages and Systems, 2006, 28, 619-695.	1.7	163
67	Running the manual. , 2006, , .		24
68	Verified Java Bytecode Verification (Verified Java Bytecode Verification). IT - Information Technology, 2005, 47, 107-110.	0.6	1
69	A Unified Memory Model for Pointers. Lecture Notes in Computer Science, 2005, , 474-488.	1.0	12
70	Verified bytecode verification and type-certifying compilation. The Journal of Logic and Algebraic Programming, 2004, 58, 27-60.	1.4	6
71	Verified bytecode verifiers. Theoretical Computer Science, 2003, 298, 583-626.	0.5	70
72	Verified Bytecode Subroutines. Lecture Notes in Computer Science, 2003, , 55-70.	1.0	2

#	Article	IF	Citations
73	Verified lightweight bytecode verification. Concurrency Computation Practice and Experience, 2001, 13, 1133-1151.	1.4	25
74	Reasoning about Translation Lookaside Buffers. , 0, , .		1