

Contents

1	Introduction	1
1.1	Motivation	1
1.2	Roadmap	2
1.3	Organisation	3
	References	3
2	Preliminaries	5
2.1	Verifiable Computation	5
2.2	Properties of Verifiable Computing Schemes	6
2.2.1	Security	7
2.2.2	Privacy	8
2.2.3	Efficiency	10
	References	10
3	Proof and Argument Based Verifiable Computing	13
3.1	Introduction to Proof and Argument Based Approaches	13
3.2	Interactive Proof Based Approaches	14
3.2.1	Verifiable Computation with Massively Parallel Interactive Proofs	15
3.2.2	Allspice: A Hybrid Architecture for Interactive Verifiable Computation	15
3.3	Interactive Argument Based Approaches	16
3.3.1	Pepper: Making Argument Systems for Outsourced Computation Practical (Sometimes)	17
3.3.2	Ginger: Taking Proof-Based Verified Computation a Few Steps Closer to Practicality	17
3.3.3	Zaatar: Resolving the Conflict Between Generality and Plausibility in Verified Computation	17
3.3.4	Pantry: Verifying Computations with State	17
3.3.5	River: Verifiable Computation with Reduced Informational Costs and Computational Costs	18

3.4	Non-interactive Argument Based Approaches.....	18
3.4.1	Pinocchio: Nearly Practical Verifiable Computation.....	19
3.4.2	Geppetto: Versatile Verifiable Computation	19
3.4.3	SNARKs for C: Verifying Program Executions Succinctly and in Zero Knowledge	20
3.4.4	Succinct Non-interactive Zero Knowledge for a von Neumann Architecture	20
3.4.5	Buffet: Efficient RAM and Control Flow in Verifiable Outsourced Computation	20
3.4.6	ADSNARK: Nearly Practical and Privacy-Preserving Proofs on Authenticated Data	20
3.4.7	Block Programs: Improving Efficiency of Verifiable Computation for Circuits with Repeated Substructures.....	21
	References	21
4	Verifiable Computing from Fully Homomorphic Encryption.....	23
4.1	Definitions for Fully Homomorphic Encryption	23
4.2	Verifiable Computing Schemes Based on FHE.....	24
4.2.1	Non-interactive Verifiable Computing: Outsourcing Computation to Untrusted Workers.....	24
4.2.2	Improved Delegation of Computation Using Fully Homomorphic Encryption.....	25
	References	25
5	Homomorphic Authenticators	27
5.1	Definitions for Homomorphic Authenticators	27
5.2	Verifiable Computing Schemes Based on MACs.....	30
5.2.1	Verifiable Delegation of Computation on Outsourced Data	30
5.2.2	Generalized Homomorphic MACs with Efficient Verification	30
5.2.3	Efficiently Verifiable Computation on Encrypted Data	31
5.3	Signature Based Verifiable Computing on Linear Functions	31
5.3.1	Programmable Hash Functions Go Private: Constructions and Applications to (Homomorphic) Signatures with Shorter Public Keys	31
5.4	Signature Based Verifiable Computing for Polynomial Functions	32
5.4.1	Homomorphic Signatures with Efficient Verification for Polynomial Functions.....	32
5.4.2	Algebraic (Trapdoor) One-Way Functions and Their Applications	32
5.5	Signature Based Verifiable Computing Using Homomorphic Encryption	33
	References	34

- 6 Verifiable Computing Frameworks from Functional Encryption and Functional Signatures** 37
 - 6.1 Verifiable Computation from Functional Encryption 37
 - 6.1.1 Verifiable Computation from Attribute Based Encryption 38
 - 6.1.2 Delegatable Homomorphic Encryption with Applications to Secure Outsourcing of Computation 38
 - 6.2 Verifiable Computation from Functional Signatures 39
 - 6.2.1 Functional Signatures and Pseudorandom Functions 39
 - References 41
- 7 Verifiable Computing for Specific Applications** 43
 - 7.1 From Secrecy to Soundness: Efficient Verification via Secure Computation 43
 - 7.2 Signatures of Correct Computation 44
 - 7.3 Efficient Techniques for Publicly Verifiable Delegation of Computation 44
 - 7.4 Efficient Computation Outsourcing for Inverting a Class of Homomorphic Functions 45
 - 7.5 Secure Delegation of Elliptic-Curve Pairing 45
 - 7.6 Efficiently Verifiable Computation on Encrypted Data 46
 - 7.7 Verifiable Delegation of Computation over Large Datasets 46
 - 7.8 Batch Verifiable Computation with Public Verifiability for Outsourcing Polynomials and Matrix Computations 46
 - 7.9 TrueSet: Nearly Practical Verifiable Set Computations 46
 - References 47
- 8 Analysis of the State of the Art** 49
 - 8.1 Security, Privacy, and Efficiency 49
 - 8.2 Long-Term Privacy 53
 - 8.3 Implementations 54
 - References 54
- 9 Conclusion** 57
 - References 58
- A Assumptions** 59
 - References 62



<http://www.springer.com/978-3-319-53797-9>

Privately and Publicly Verifiable Computing Techniques

A Survey

Demirel, D.; Schabhüser, L.; Buchmann, J.

2017, XII, 64 p., Softcover

ISBN: 978-3-319-53797-9