Crypto 2001

1. Universally Composable Commitments, https://link.springer.com/chapter/10.1007/3-540-44647-8_2
    a. F_com: The Ideal Commitment functionality for a single commitment
    b. F_mcom: The Ideal Commitment functionality for multiple commitments
    c. F_zk: The Zero-Knowledge functionality

Crypto 2002

1. Separating Random Oracle Proofs from Complexity Theoretic Proofs: The Non-committing Encryption Case
    a. F_ncs: non-committing encryption
2. A Threshold Pseudorandom Function Construction and Its Applications
    a. F_{F,c} (408): evaluation of function family
    b. F_{tsig,c} (409): threshold signatures
    c. F_{ba,t} (409): byzantine agreement
3. Security Analysis of IKE's Signature-Based Key-Exchange Protocol
    a. Not sure of this one, not obvious functionalities used
4. Perfect Hiding and Perfect Binding Universally Composable Commitment Schemes with Constant Expansion Factor
    a. F_hcom (606): homomorphic commitments (multi party)
    b. F^{r}_{zk} (609): zero knowledge for a binary relation r

Crypto 2003

1. Universally Composable Efficient Multiparty Computation from Threshold Homomorphic Encryption
    a. F_{abb} (271): arithmetic black box
    b. Might have missed one
2. Universal Composition with Joint State
    a. F_comm (287): taken from UC paper
    b. F^{t}_{com} (289): commitment parameterized by security parameter t
    c. ^F_crs: multi session extension of crs
    d. ^F^{r}_{zk}: multi session extension of F_zl
3. On Deniability in the Common Reference String and Random Oracle Model
    a. Really just proofs of possibility
4. Relaxing Chosen-Ciphertext Security
    a. F_pke (573): ideal public key encryption functionality
    b. F_rpke (579,): public key encryption encryption scheme for RCCA security, just an extenson of F_pke to explicitly allow the environment to generate ciphertexts that decrypt to the same value as a given ciphertext

Crypto 2004

1. Zero-Knowledge Proofs and String Commitments Withstanding Quantum Attacks
    a. Seems to use something close to UC, with quantum zero knowledge functionalities
    b. Commitment scheme as well
2. Adaptively Secure Feldman VSS and Applications to Universally-Composable Threshold Cryptography
    a. F_smt (331): referenced in another paper
    b. F_ssmt (331): spooled smt deliver one message when another is sent
    c. F^{com_k}_{vss} (331): committed vss protocol with commitment scheme comm_k,
    d. F_ssmt_wo: not important used in hybrid model
    e. F_hgen (338): commitment-key generation functionality
    f. F_tsig (341): threshold schnorr signature

Crypto 2005

1. A Formal Treatment of Onion Routing
    a. F_onionrouting (182): text only description of it
2. Secure Computation Without Authentication (interesting, check it out)
    a. sF (380): split version of some functionality, initializtion phase where honest parties are set and required to be disjoint (check it out)
    b. F_auth (381): authentication functionality → F_SA split authenication
3. Constant-Round Multiparty Computation Using
    a. Secure composition of low-degree polynomials (393)

Crypto 2006

1. Round-Optimal Composable Blind Signatures in the Common Reference String Model
    a. F_bsig (79): blind signature functionality
2. On Signatures of Knowledge
    a. F_sig (94): just canetti's signature functionality
    b. F_SOK (95): signature of knowledge for a witness x
    c. F_sok(F_0) (104): signature of knowledge with verification functionality F_0
3. Mitigating Dictionary Attacks on Password-Protected Local Storage
    a. F_pkgr(D, p, m) (184): password key generation and recovery functionality parameterized with dictionary D, maximum number p of password queries, and length m of the generated key
4. Receipt-Free Universally-Verifiable Voting with Everlasting Privacy
    a. Voting scheme (390) - functions Vote and Change vote
5. On Expected Constant-Round Protocols for Byzantine Agreement

    a. Only does sequential composition not UC

Crypto 2007

1. Universally-Composable Two-Party Computation in Two Rounds
    a. F_zk from CLOS02
    b. F two part computation in F_zk-hybrid world (128)
    c. 130 two round two output in hybrid
2. Cryptography in the Multi-string Model
    a. F_mcrs (347): ideal multi-string generator
    b. F_com (348): commitment functionality
3. A Tight High-Order Entropic Quantum Uncertainty Relation with Applications
    a. none

Crypto 2008

1. Scalable Multiparty Computation with Nearly Optimal Work and Resilience
    a. F_monochrom (257): generate a random sharing [b]_d where block b is (0,...,0) or (1,...,1) with prob ½
    b. F_CP (261): Computes an arithmetic circuit A
2. Cryptographic Complexity of Multi-Party Computation Problems: Classifications and Separations
    a. Splitability of 2-party functionalities
    b. F_pvt (279): private channel functionality
3. Collusion-Free Protocols in the Mediated Model
    a. Uses previous zk functionalities
4. Efficient Constructions of Composable Commitments and Zero-Knowledge Proofs
    a. G_acrs (533): used from guc paper
    b. F_zk^r (536): zero knowledge over relation R
    c. F_com (539)
5. Commitments and Zero-Knowledge Proofs Composable Oblivious Transfer
    a. F_ot (571): oblivious transfer → multis session extended (with JUC?)
6. Founding Cryptography on Oblivious Transfer –Efficiently
    a. More just about OT didn't see an Fs

Crypto 2009

1. Improving the Security of Quantum Protocols via Commit-and-Open
    a. F_ID (435): ideal password-based identification functionality, checks if input from server/user are equal
2. Somewhat Non-comm itting Encryption andEfficient Adaptively Secure Oblivious Transfer

    a. F_SC^N (522): parameterized secure-channel ideal functionality, parameterized by non-information oracle N which gets values from messages and outputs some side info to adversary
    b. F_SFE^f (527): two-party secure function evaluation for initiator I and responder R with specific input domain
3. Smooth Projective Hashing for Conditionally Extractable Commitments
    a. None

Crypto 2010

1. Credential Authenticated Identification and Key Exchange
    a. F_ach (261): authenticated channels sender P receiver Q, adversary gets whole message
    b. F_sch (261): secure channels
    c. F_caid (262): credential authenticated identification
    d. F_cake (263): credential authenticated key exchange
    e. F_ske (266): split key exchange (split like above)
    f. F_ezk (268): enhanced zero knowledge, some information is leaked to the adversary about the input x
2. Password-Authenticated Session-Key Generation on the Internet in the Plain Model
    a. Password authenticated key exchange as a two-party functionality F (282)
3. Instantiability of RSA-OAEP under Chosen-Plaintext Attack
4. Universally Composable Incoercibility
    a. Voting functionality (422)
5.
    a. Coin Toss With Dealer with Dealer (550)
    b. Multishare Gen (553): computes the bits for the underlying sets and the signed shares for the inner and outer secret-sharing schemes
    c. Protocl multiparty coin toss (554)
    d. Functionality Reconstruction (555)
6. Multiparty Computation for Dishonest Majority: From Passive to Active Security at Low Cost
    a. F_amc (564): arithmetic MPC, add and multiply
    b. F_rand (564): generate a key for a double-trapdoor homomorphic commitment scheme
7. Secure Multiparty Computation with Minimal Interaction
    a. None
8. A Zero-One Law for Cryptographic Complexity with Respect to Computational UC Security
    a. More about functionalities, reactive functionalities
9. Secure Two-Party Quantum Evaluation of Unitaries against Specious Adversaries
    a. Some parameters are ideal functionalities, look at more closely
10. On the Efficiency of Classical and Quantum Oblivious Transfer Reductions

    a. F_mcom (717)
11.

Crypto 2011

1. Physically Uncloneable Functions in the Universal Composition Framework
    a. F_puf (60):
    b. F_OT (62): adopted from elsewhere
    c. F_com (64): elsewhere
    d. F_ke (67): elsewhere
2. The IPS Compiler: Optimizations, Variants and Concrete Efficiency
    a. Multisender k-out-of-n oblivious transfer F^k_n
3. Classical Cryptographic Protocols in a Quantum World
    a. G_zk (421): zero knowledge functionality
    b. Quantum stand-alone emulation
    c. Check on more
4. Constant-Rate Oblivious Transfer from Noisy Channels

Crypto 2012

1. Must You Know the Code of f to Securely Compute f ?
    a. Hiding functionalities
    b. F_m (95): simulate PPT machine M
    c. F_pi
2. Adaptively Secure Multi-Party Computation with Dishonest Majority
    a. F_n-crs (120): look into this
3. Collusion-Preserving Computation, interesting look at this
4. Secure Database Commitments and Universal
    a. F_DbCom (249): database commitment functionality
5. On the Security of TLS-DHE in the Standard Model
6. Universal Composability from Essentially Any Trusted Setup
7. Impossibility Results for Static Input Secure Computation
    a. F_universal (438):
8. New Impossibility Results for Concurrent Composition and a Non-interactive Completeness Theorem for Secure Computation
    a. F_ot
9. Multiparty Computation from Somewhat Homomorphic Encryption
    a. F_keyget (653): distributed keygen
    b. F_keygendec (654): keygen and decryption
10. A New Approach to Practical Active-Secure Two-Party Computation
    a. F_2pc (688): boolean two party computation
    b. F_deal (688): dealing preprocessed values
    c. F_2pc in F_deal-hybrid world (689)
11. Actively Secure Two-Party Evaluation Actively Secure Two-Party Evaluation

a. Not the same ideal functionality?

Crypto 2013

1. On Fair Exchange, Fair Coins and Fair Sampling
    a. Uses F_exch: fair exchange of a single bit
    b. F_coin: obtain a common coin, fair coin toss
    c. F_rOT: sampling of a random instance of oblivious transfer:
        i. Alice gets pair of random bits (x0, x1) and bob gets (b,xb) where b is a random bit
2. Secure Computation against Adaptive Auxiliary Information
    a. None
3. A Dynamic Tradeoff between Active and Passive Corruptions in Secure Multi-Party Computation
    a. Ideal functionality for non-reactive MPC (208)
4. What Information Is Leaked under Concurrent Composition?
    a. Just results about composition
5. Quantum One-Time Programs
    a. F_^OTP_PHI (348): one-time program that computes some function and deletes any trace of the instance of the quantum channel
    b. F_^delegated_PHI (359): program for computation delegation
6. Everlasting Multi-party Computation
    a. The only difference between the definition from [10] and ours is that we allow the environment to output a quantum state and that we require that state to be trace-indistinguishable between real and ideal model.
    b. No new functionalities?

Crypto 2014

1. Leakage-Tolerant Computation with Input-Independent Preprocessing
    a. F^f_2LTC-AUX (160): 2-party leaky ideal functionality computing f with auxiliary inputs
    b. F_LSC (160): secure communication functionality
    c. F_LFS (160): input-dependent leakage-free preprocessing functionality which provides the initial states for all parties
2. Memento: How to Reconstruct Your Secrets from a Single Password
    a. F_TPASS(t,n) (261): t-out-of- n TPASS functionality
        i. Threshold password-authenticated secret sharing
3. Physical Zero-Knowledge Proofs of Physical Properties
    a. F_ZK^pi (317): running with parties Prover, Verifier, and an oracle F_C that compiles the ideal functionalities for a collection C of physical operations in the real world
    b. F_ZK^BBE (333): zero knowledge proof of bins and balls equality with soundness error delta

4. Secure Multi-Party Computation with Identifiable Abort
    a. F_corr^D (373): sampling functionality, on default input from any party, samples R from D and distributes it to the parties
5. Feasibility and Infeasibility of Secure Computation with Malicious PUFs
    a. F_PUF: physically unclonable functions, seen before
6. How to Use Bitcoin to Design Fair Protocols
    a. F_cr (429): claim and refund functionality where sender can conditionally send coins to a receiver, receiver must act within a time bound
    b. F_f* (430): secure computation with penalties
    c. F_llot* (431): secure lottery with penalties
7. Dishonest Majority Multi-Party Computation for Binary Circuits
    a. F_online (501): online phase of secure computation
    b. F_prep (502): preprocessing functionality
    c. F_aBit (503): two-party bit authentication
    d. F_bootstrap (505): the outputs of F Bootstrap as the shares of scalar products $x \cdot \delta$, where one party P i chooses the scalar (bit) x
    e.

Crypto 2015

1. A Simpler Variant of Universally Composable Security for Standard Multiparty Computation
    a. F_com in SUC (18)
2. Concurrent Secure Computation via Non-Black Box Simulation
    a. F_com1,com2 (38): 2 non-interactive perfectly binding, concurrently secure computation
3. Concurrent Secure Computation with Optimal Query Complexity
    a. No functionalities suggested
4. Constant-Round MPC with Fairness and Guarantee of Output Delivery
    a. No functionality suggested
5. Efficient Zero-Knowledge Proofs of Non-algebraic Statements with Sublinear Amortized Cost
    a. F_zk^r: zk for relation R
    b. F_com (156): looks same as before
    c. F_otc (156): committing oblivious transfer
    d. F_Aut (156): authentiicated array access
    e. F_cpfe (157): committing private function eval
    f. F_init (164): initializing an ORAM program along with wire labels
6. Cryptography with One-Way Communication
    a. C_rot (199): random oblivious transfer channel
    b. C_BEC : binary erasure channel
    c. C_BSC: ?
7. Efficient Constant Round Multi-party Computation Combining BMR and SPDZ

    a. F_sfe (324): secure function evaluation, input is binary circuit C for function f
    b. F_MPC (325): generic reactive MPC functionality. five externally exposed commands Initialize, InputData, Add, Multiply, and Output, and one internal subroutine Wait
    c. F_offline (326): offline phase and preprocessing
    d. F_SPDZ (330): SPDZ functionality
8. PoW-Based Distributed Cryptography with No Trusted Setup
    a. F_syn^pi,pi_A,theta (388): exponent model described on 387, synchronizing parties, tracks computational resources,
    b. F_bc^T (390): T is the bound on number of fake identities the adversary can create, broadcast functionality
9. Large-Scale Secure Computation: Multi-party Computation for (Parallel) RAM Programs
    a. F_PRAMs (755): secure computation of a sequence of adaptively chosen PRAMs on parties' inputs
10. Incoercible Multi-party Computation and Universally Composable Receipt-Free Voting
    a. T_ThEnc (P) (777): threshold encryption token functionality

Crypto 2016

1. Network-Hiding Communication and Applications to Multi-party Protocols
    a. F_Ng (345): network functionality that knows every party's neighborhood
    b. Topology hiding secure multi party computation
    c. F_abc (361): anonymous broadcats
2. On the Communication Required for Unconditionally Secure Multiplication
    a. No sign of NEW functionality
3. Rate-1, Linear Time and Additively Homomorphic UC Commitments
    a. F_hcom (185): homomorphic commitments
    b. F_rot (186): random oblivious transfer
    c. F_ot (203): oblivious transfer
4. UC Commitments for Modular Protocol Design and Applications
    a. F_nic (216): non-interactive commitments
    b. F_enic (216): extractable non-interactive commitments
    c. F_rev (225): revocation
    d. F_at (231): anonymous attribute tokens
    e. F_tr (234): anonymous revocable attribute tokens
5. Probabilistic Termination and Composability of Cryptographic Protocols
    a. F^{f,l})_CSF (250): canonical synchronous functionality with function f and leakage l
    b. Uses F_ba , F_smt, F_sfe,

    c. W^D_strict(F) (255): strict-wrapper generates a trace from distribution, according to that defines termination round and outputs result of F to adversary
    d. W^D_flex (F) (255): adversary can instruct wrapper to deliver output to each party at any round.
    e. W^{D,c}_{sl-strict](F) (257): slack tolerance
    f. W^{D,c}_{sl-flex}(F) (260): slack with flex wrapper
6. Efficient Zero-Knowledge Proof of Algebraic and Non-Algebraic Statements
    a. F_cot (505): sender commits to a seed, oblivious transfer
    b. F_com,f (509): commitment revealed if function f passes on input
    c. F_hash (519): commit to message and it's hash
    d. F_eq (521): equality of committed values in different groups . The goal is to prove that the value committed to in different prime groups of size p and q are the same.

Crypto 2017

1. The TinyTable Protocol for 2-Party Secure Computation
    a. F^{pre}_{sem} (171): preprocessing phase in semi-honest case
    b. F^{pre}_{mal} (172): preprocessing, malicious security
    c. F_sfe (172): secure function eval, malicious security
    d. F^{pre}_{freeXOR,sem} (176): preprocessing, semi honest, free XOR
    e. F^{pre}_{MiniMac} (184): preprocessing using MiniMac protocol
    f. F_table (184): giving on-line access to tables
2. The Bitcoin Backbone Protocol with Chains of Variable Difficulty
    a. F_RO (296): random oracle functionality
3. Bitcoin as a Transaction Ledger: A Composable Treatment
    a. G_ledger (339): bitcoin like ledger
    b. F_^{delta,p_h,p_a,....}_{stx} (347): parties submit ledger states which are accepted with some probability and multicast for everyone
4. Ouroboros: A Provably Secure Proof-of-Stake Blockchain Protocol
    a. F^{D,f}_LS (370): leader selection with distribution D, function F to select
    b. "" (379): leader election but now with trusted beacon, functionality to now provide randomness and auxiliary information F LS for the leader election process throughout the epochs
5. The Price of Low Communication in Secure Multi-party Computation
6. Topology-Hiding Computation on All Graphs
    a. F_graph (453): gets network graph then tells all parties their neighbors, then acts as ideal channel for communicating with neighbors
    b. F_broadcast (453): sends bit to all parties
7. Fast Secure Two-Party ECDSA Signing
    a. F_com (618): commitment

b. F^R_zk (618):
c. F^R_com-zk (619)
d. F_ECDSA (641): works with 2 parties to sign messages

Crypto 2018

1. Anonymous Attestation with Subverted TPMs
    a. F_pdaa (434): DAA-with-optimal-privacy functionality, direct anonymous attestation (DAA)
    b. F^l_daa (437): above is strengthened version of this (proposed somewhere else)
2. Fast Distributed RSA Key Generation
    a. F_ot (336): random ot
    b. F_ct (337): coin tossing
    c. F^M_zk (337): language verification functuon M_L
    d. F^f_2PC (338): 2 party computation
    e. F_RSA-semi (343): generating a shared RSA key semi-honestly
    f. F_rsa (352): generating shared RSA-ley
3. Round-Optimal Secure Multi-Party Computation
    a. F^A_mult (500): additively corruptable 3-bit multiplication
4. Multi-Theorem Preprocessing NIZKs from Lattices
    a. F_bhs (753): blind homomorphic signatures
5. SPDZ 2 k : Efficient MPC mod 2 k for Dishonest Majority
    a. F_rot (774)
    b. F_mac (776): generating shares of global MAC key, distributing shares of inputs and tags
    c. F_Prep (782): preprocessing is F_mac with additional functions
    d. F_Online (782): online phase
    e. F^s_vOLE (784): Random vector oblivious linear function evaluation functionality over Z_2^k+s
    f. F_triple
6. Yet Another Compiler for Active Security
    a. F_bcast (805): broadcast
    b. F_cflip (805): coin flip
    c. F_triple (806): triple generation
7. TinyKeys: A New Approach to Efficient Multi-Party Computation
    a. F^r_zero (P) (11): random zero sharing functionality
    b. F^r,l_d-rot (13): oblivious transfer on random, correlated strings
    c. F^r,l_leaky-2-mult (14): eaky secret-shared two-party bit multiplication
    d. F^r_triple (16): Multiplication triple generation
    e. F^{l_bmr}_Preprocessing (20): Multi-party garbling
    f. F_bitxbit (23): Secret-shared bit multiplication

    g. F^{l_BMR}_BitString (23): Secret-shared bit/string multiplication
8. Fast Large-Scale Honest-Majority MPC for Malicious Adversaries
    a. F_rand (40): generating random shares
    b. F_input (40): sharing inputs
    c. F_mult (42): Secure Mult. Up To Additive Attack
    d. F_checkzero (42): Checking Equality to 0
    e. F_product (53): Sum-of-Products Up To Additive Attacks
9. Must the Communication Graph of MPC Protocols be an Expander?
    a. F_elect-share (258):
    b. F_reconstruct-compute
    c. F_out-dist
10. Two-Round Multiparty Secure Computation
    a. F_f (285): function eval
    b. F_zk (291): special zero knowledge
11. Optimizing Authenticated Garbling for Faster Secure Two-Party Computation
    a. F_abit (368); authenticated-bits
    b. F_pre (369): preprocessing for some fixed circuit
    c. F_land (382): computing leaky AND triple

EuroCrypt 2003

1. On the Limitations of Universally Composable Two-Party Computation without Set-up Assumptions
    a. About UC
2. Strengthening Zero-Knowledge Protocols Using Signatures
    a. F_zk (203): multi session zero knowledge for relation R
3. Round Efficiency of Multi-party Computation with a Dishonest Majority
    a. None

EuroCrypt 2004

1. On Simulation-Sound Trapdoor Commitments
    a. F_zk (407): multi session zkp

EuroCrypt 2005

1. Universally Composable Password-Based Key Exchange
    a. F_ke (421): authenticated key exchange
    b. F_pwke (422): password based key exchange
    c. F_pwsc (430): password based secure channels

EuroCrypt 2006

1. Polling with Physical Envelopes: A Rigorous Analysis of a Human-Centric Protocol
    a. Crrt cryptographic randomized response technique
    b. Strong p-crrt (93): deniability and bounded bias requirements
    c. Pollster immnue p-crrt (93): where a malicious pollster cannot learn more than an honest pollster about the responder's secret bit
2. Perfect Non-interactive Zero Knowledge for NP
    a. F_nizk (351): non interaction zkp
3. On the (Im-)Possibility of Extending Coin Toss
    a. CT_n (509): n-bit coin toss
4. Information-Theoretic Conditions for Two-Party Secure Function Evaluation
    a. F_ot (548)
5. Simplified Threshold RSA with Adaptive and Proactive Security
    a. F_thsig (598): canetti's signature adapted for threshold
    b. F_keygen (600):

EuroCrypt 2007

1. Efficient Two-Party Secure Computation on Committed Inputs
    a. F_cot (117): commited oblivious transfer on strings
    b. F_2PC (121): 2-party secure computation on commited inputs
2. Universally Composable Multi-party Computation Using Tamper-Proof Hardware
    a. F_wrap (132): wrapper for secure hardware, party takes code seals it in the hardware and gives an access token to another party
    b. F_mcom (134): multiple commitment
3. Non-interactive Proofs for Integer Multiplication
    a. F_vss (428):
    b. F_ab=c

EuroCrypt 2008

1. Efficient Two Party and Multi Party Computation Against Covert Adversaries
    a. None
2. David and Goliath Commitments: UC Computation for Asymmetric Parties
    a. F_com, F_bcom, F_mcom (543): bounded commitment
    b. F_wrap
3. New Constructions for UC Secure Computation Using Tamper-Proof Hardware
    a. F_wrap (561)
    b. F_adv (561): adversarial functionality used to create tokens for F_wrap

EuroCrypt 2009

EuroCrypt 2010

1. Partial Fairness in Secure Two-Party Computation
    a. ShareGen_r (837): a randomized functionality that returns shares to each part
    b. ShareGen_p,r (841): same but with probabilistic security
2. Adaptively Secure Broadcast
    a. F_bc (1145): broadcast
    b. F_ubc (1146): unfair broadcast, can corrupt sender and modify the message
3. Quantum Multi-party Computation
    a. Just new framework

EuroCrypt 2011

1. Semi-homomorphic Encryption and Multiparty Computation
    a. F_trip (182): making singles [a] and triples [a], [b], [c]
    b. F_ampc (183): arithmetic mpc
2. Efficient Non-interactive Secure Computation
    a. No new functionalities
3. Towards a Game Theoretic View of Secure Computation
    a. f^alpha (443): compute f with security-with-abort
4. Highly-Efficient Universally-Composable Commitments Based on the DDH Assumption
    a. F_mcom (450): realized in this paper
5. Concurrent Composition in the Bounded Quantum Storage Model
    a. About UC, no new functionality

EuroCrypt 2012

1. Concurrently Secure Computation in Constant Rounds
    a. No functionalities

EuroCrypt 2013

1. MiniLEGO: Efficient Secure Two-Party Computation from General Assumptions
    a. F_sfe (540): secure function evaluation for two parties
    b. F_com (543): commitment scheme
    c. F_wcom (551): wildcard commitments
2. Multi-party Computation of Polynomials and Branching Programs without Simultaneous Interaction
3. On Concurrently Secure Computation in the Multiple Ideal Query Model

    a. F_token (694): token that is used to run obfuscated program
4. Universally Composable Secure Computation with (Malicious) Physically Uncloneable Functions
    a. F_PUF (710): functionality for malitious PUFs

EuroCrypt 2014

1. Non-Interactive Secure Computation Based on Cut-and-Choose
    a. MS-NISC (397):
2. On the Complexity of UC Commitments
    a. F_mcom (681)
    b. F_ot^N (682): 1-out-of-N oblivious transfer
    c. F_otr^d (682): Rabin-OT with noise rate delta
3. Universally Composable Symbolic Analysis for Two-Party Protocols Based on Homomorphic Encryption
    a. F_aux is used?

EuroCrypt 2015

1. How to Efficiently Evaluate RAM Programs with Malicious Security
    a. F_xcom (709): xor-homomorphic commitment functionaity
2. Leakage-Resilient Circuits Revisited
    a. F_sampling^delta (145): sampling correlated randomness
3. Cryptographic Reverse Firewalls

EuroCrypt 2016

1. Secure Computation from Elastic Noisy Channels
    a. F_com (204): commitments from (a,B)-BCS for 0<B<a<1
    b. F_ot (207): parameterized by probabiliy function for setting bit b
2. Fair and Robust Multi-party Computation Using a Global Transaction Ledger
    a. G_clock (713)
    b. G_ledger (715)
    c. W_Q,G (F): Q-fairness wrapper
3. Two Round Multiparty Computation via Multi-key FHE
    a. A basic MPC protocol for f secure against N − 1 corruptions (758)

EuroCrypt 2017

1. Improved Private Set Intersection Against Malicious Adversaries
    a. F_ot (238): 1-out-of-2 or
    b. F_psi (238): private set intersection

2. Formal Abstractions for Attested Execution Secure Processors
    a. F_att (268): global functionality modeling sgx-like secure processor
3. Unconditional UC-Secure Computation with (Stronger-Malicious) PUFs
    a. F_puf (391)
    b. F_hpuf (405): honest pufs
    c. F_mpuf (406): malicious pufs
    d. F_e-puf (407): malicious pufs that may encapsulate pufs
4. Sublinear Zero-Knowledge Arguments for RAM Programs
    a. G_gRO (510): parametrized by set of functionalities
    b. F_zk (510):
    c. F^{C1,C2}_check (510): input 2 check circuits that each take one input
5. High-Throughput Secure Three-Party Computation
    a. F^1_cr, F^2_cr (233): correlated randomness
    b. F_rand (235): generating shares of a random value
    c. F_coin (237):
    d. F_reconst (238): secure reconstruction
    e. F_share (239): sharing a secret
    f. F_triples (243): generating multiplication triples
6. Decentralized Anonymous Micropayments
7. Faster Secure Two-Party Computation in the Single-Execution Setting
    a. F_ot (404): oblivious transfer
    b. F_cOR (404): commiing oblivious transfer
8. Non-interactive Secure 2PC in the Offline/Online and Batch Settings
    a. F_b2pc (434): batch 2PC
    b. F_ “ (449): offline/online 2PC

EuroCrypt 2018

1. The Wonderful World of Global Random Oracles
    a. F_sRO (287): strict global random oracle functionality G sRO that does not give any extra power to anyone (mentioned but not defined by Canetti)
    b. F_sig (290): from canetto
    c. F^L_PKE (293): public-key encryption with leakage function L
    d. F_pRO (295): programmable global random oracle
2. But Why Does It Work? A rational protocol design treatment of bitcoin
    a. F_ledger weakened version
3. Ouroboros Praos
    a. F_kes (76): key evolving signature schemes, forward security with erasures
    b. F_vrf (78): generates new verification key that is used to label a table
    c. F_init (80): used by all parties to send messages and keep synchronized with a global clock
    d. F_^{t,r}_RLB (93): dynamic stake case with resettable leaky beacon

4. Efficient Maliciously Secure Multiparty Computation for RAM
    a. F_ram (102): computes program from inputs
    b. F_uap (103): unprotected access pattern security
    c. F_bmr (106):
5. Overdrive: Making SPDZ Great Again
    a. F^S_{SKPoK} (164): proof of knowledge ciphertext
    b. F_[[]] (170): commits the parties to secret sharings and that provides the secure computation
    c. F_triple (174): random triple generation
    d. F^S_{gZKPoK} (180): global proof of knowledge of ciphertext
6. Exploring the Boundaries of Topology-Hiding Computation
    a. F_graph (304):
    b. F_graph-failstop (306):
    c. F_fs-broadcast (318)
7. Fuzzy Password-Authenticated Key Exchange
    a. F_pake (399)
    b. F^P_RFE (405): randomized fuzzy equality
    c. sF_^P_RFE (409):
    d. F_iPAKE (413): implicit-only PAKE
8. OPAQUE: An Asymmetric PAKE Protocol Secure
    a. F_aPAKE , F_saPAKE
       F_OPRF (464): obliious pseudorandom function with adaptive compromise
    b. F_AKE-KCI (469):

NDSS 2014

1. Decentralized Anonymous Credentials
    a. F_dac : basic distributed anonymous credential system

NDSS 2017

1. TumbleBit: An Untrusted Bitcoin-Compatible Anonymous Payment Hub
    b. F_fair-RSA
2. Fast Actively Secure OT Extension for Short Secrets
    a. F^(n,m,l)_OT : n choose 1 OT^l_m
    b. F_coin : generating random common coins
3. SilentWhispers: Enforcing Security and Privacy in Decentralized Credit Networks
    a. F_rout : routing algorithm
    b. F_pay: pay operation in credit network
    c. F_acc : accountability mechanism
4. HOP: Hardware makes Obfuscation Practical
    a. F^RAM_obf : obfuscated ram
    b. F_token : secure hardware token

5. Constant Round Maliciously Secure 2PC with Function-independent Preprocessing using LEGO
    a. tinyLEGO in (F_hcom , F_d_rot)-hybrid world

USENIX 2014

1. Ad-Hoc Secure Two-Party Computation on Mobile Devices using Hardware Tokens
    a.

USENIX 2015

1. The Pythia PRF Service
2. M 2 R: Enabling Stronger Privacy in MapReduce Computation
    a.

USENIX 2016

1. Faster Malicious 2-party Secure Computation with Online/Offline Dual Execution
    a. F_multi-sfe : (ε-leaking) secure function evaluation

USENIX 2017

1. MCMix: Anonymous Messaging via Secure Multiparty Computation
    a. F_mpc : mpc functionality for a family of programs {f_z}_z

USENIX 2018

1. Simple Password-Hardened Encryption Services
2. GAZELLE: A Low Latency Framework for Secure Neural Network Inference
3. Enter the Hydra: Towards Principled Bug Bounties and Exploit-Resistant Smart Contracts
    a. F_withold : captures front-running, but is far stronger than previous models (e.g., Hawk [35])
4. Practical Accountability of Secret Processes
    a. F_ledger : just described in brief with high level text

Oakland 2014:

1. W YSTERIA : A Programming Language for Generic, Mixed-Mode Multiparty Computations
2. Formal Analysis of Chaumian Mix Nets with Randomized Partial Checking
3. Dynamic Searchable Encryption via Blind Storage

    a. F_store : blind storage with leakage
    b. F_sse : searchable symmetric encryption
4. Automating Efficient RAM-Model Secure Computation
5. An Expressive Model for the Web Infrastructure: Definition and Application to the BrowserID SSO System
6. ANONIZE: A Large-Scale Anonymous Survey System
7. Blind Seer: A Scalable Private DBMS
    a. F_db captures the privacy requirement of a database management system in which each query is handled deter- ministically
    b.

Oakland 2015

1. SoK: A comprehensive analysis of game-based ballot privacy definitions
    a. F_voting
2. Secure Sampling of Public Parameters for Succinct Zero Knowledge Proofs
3. Malicious-Client Security in Blind Seer: A Scalable Private DBMS
    a. F_db : doesnt appear to be pecified
4. ObliVM: A Programming Framework for
    a.

Oakland 2016

1. Hawk: The Blockchain Model of Cryptography and Privacy-Preserving Smart Contracts
    a. F_idealcash, ledger, etc, you know what all there is
    b. F_F_
    c.

Oakland 2017

1. One TPM to Bind Them All: Fixing TPM 2.0 for Provably Secure Anonymous Attestation
    a. F_pdaa+ : ideal DAA functionality with strong privacy
2. SecureML: A System for Scalable Privacy-Preserving Machine Learning
    a. F_ot
    b. F_ml
3. Machine-Checked Proofs of Privacy for Electronic Voting Protocols
4. A Framework for Universally Composable Diffie-Hellman Key Exchange
    a. F_crypto : cryptographic primitives
        i. i) generate symmetric keys, including pre-shared keys, ii) generate public/private keys, iii) derive symmetric keys from other symmetric keys, iv) encrypt and decrypt messages and ciphertexts, respectively (public-key encryption and both unauthenticated and authenticated

symmetric encryption are supported), v) compute and verify MACs and digital sig- natures, and vi) generate fresh nonces
    b. F^MA_key-use : key exchange mutual authentication
    c. F^UA_key-use
5. Identifying Personal DNA Methylation Profiles by Genotype Inference
    a.

Oakland 2018

1. FuturesMEX: Secure, Distributed Futures Market Exchange
    a. F_cfm : posting, cancelling, and marking to market
    b. F_: many other functionalities used here but not defined
2. Secure Two-party Threshold ECDSA from ECDSA Assumptions (wow relaly good graphics)
    a. F_ecdsa :
    b. F_sampledECDSA : this is the one that is realized (only 2 rounds needed)
    c. F_mul
    d. F_COTe : correlatev oblivious transfer extension
    e. F_sf-ot : selective failure ot
    f. F^R_zk:
3. Anonymity Trilemma: Strong Anonymity, Low Bandwidth, Low Latency—Choose Two
    a.

CCS 2014

1. A New Additive Homomorphic Encryption based on the co-ACD Problem
2. Founding Digital Currency on Secure Computation
    a. F_cs : convert sharings
    b. F_pdc : ?
3. (Nothing else) MATor(s): Monitoring the Anonymity of Tor's Path Selection
4. Multi-Ciphersuite Security of the Secure Shell (SSH) Protocol
5. Practical UC S ecurity with a Global Random Oracle
    a. F_gRO :
    b. F_tZK: zk up to transferrability
    c. F_tcom
6. Controlled Functional Encryption
7. How to Use Bitcoin to Incentivize Correct Computations
    a. F_cr* , F_f*, F*_f,leak - leak with penalty
    b. F*_ml
8. Algebraic MACs and Keyed-Verification Anonymous Credentials
    a.

CCS 2015

1. Secure Deduplication of Encrypted Data without Additional
    a. F_pake password auth blah blah
    b. F_same-input-pake
    c. F_dedup deduplicating encrypted data
2. Optimal Distributed Password Verification
    a. F_smt
    b. F_dpv :
3. DEMOS-2: Scalable E2E Verifiable Elections without Random Oracles
    a. F_k-priv :
4. Fast Non-Malleable Commitments
5. How to Use Bitcoin to Play Decentralized Poker
    a. F_cr*
    b. F*_f,d : Secure cash distribution with penalties
6. (Un)linkable Pseudonyms for Governmental Databases
    a. F_nym :
7. A Domain-Specific Language for Low-Level Secure
    a.
8. Fast and Secure Three-party Computation:
    a. F_f
9. SPRESSO: A Secure, Privacy-Respecting Single Sign-On System for the Web
10. Deniable Key Exchanges for Secure Messaging
    a. F_post-keia post-specified peer key exchange with incriminating abort
    b.

CCS 2016

1. The Ring of Gyges: Investigating the Future of Criminal Smart Contracts
    a. F_public leakage
    b. F_keytheft
2. High-Throughput Semi-Honest Secure Three-Party Computation with an Honest Majority
    a. F_mult, F_cr : corr. Randomness
3. Amortizing Secure Computation with Penalties
    a. F^ord_f : enforcing ordered deliv- ery of outpu
    b. F*_MSFE multiple sequential sfe punishents
4. Hash First, Argue Later
5. A Unilateral-to-Mutual Authentication Compiler for Key Exchange (with Applications to Client Authentication in TLS 1.3) *
6. MPC-Friendly Symmetric Key Primitives
    a. F_abb : arithmetic mpc
    b. F_abb-exp : public exponentiation

    c. F_abb-leg : Legendre symbol PRF
7. MASCOT: Faster Malicious Arithmetic Secure Computation with Oblivious Transfer
    a. F_[[]] : for authenticating, computing linear combinations of, and opening additively shared values
8. Attribute-based Key Exchange with General Policies *
    a. F_anon : anon comm
    b. F_abke : attribute based ke
    c. F_setup: setup
9. Improvements to Secure Computation with Penalties
    a. F_f* : Secure (non-reactive) computation with penalties
    b. F*_Cr , special
10. Efficient Batched Oblivious PRF with Applications to Private Set Intersection
    a. F_boprd : batched related-key oprf
    b. F_ optimization to the PSSZ PSI proto-col, written in terms of an OPRF
11. Safely Measuring Tor
    a. F_PC private count
12. Town Crier: An Authenticated Data Feed for Smart Contracts
    a. F_sgx Formal abstraction for SGX execution capturing a subset of SGX features sufficient for im- plementation of TC.
    b.

CCS 2017

1. Distributed Measurement with Private Set-Union Cardinality
    a. F_psc: private set union cardinality
    b. F_zkp-pr : re-encryption re-randomization ZKP
    c. F_zkp-s : re-encryption shuffle zkp
    d. F_zkp-dl: zero knowl- edge proof of knowledge of a discrete log
    e. F_zkp-dle : zero knowledge proof of knowledge of the equality of two discrete logs
    f. F_bc: broadcast
    g. F_sc : secure point to point communication
2. TinyOLE: Efficient Actively Secure Two-Party Computation from Oblivious Linear Function Evaluation
    a. F_deal: dealer functionality
    b. F_2pc : arithmetic two-party computation
3. A Fast and Verified Software Stack for Secure Function Evaluation
    a. Didn't see any
4. Practical Multi-party Private Set Intersection from Symmetric-Key Techniques
    a. F_psi: psi
    b. F_oprf: oblivious prf
    c. F_opprf :

5. Pool: Scalable On-Demand Secure Computation Service Against Malicious Adversaries
6. Oblivious Neural Network Predictions via MiniONN Transformations
    a. F_triplet : generate dot-product triplet
    b. F_ : oblivious activation/pooling
    c. F_ReLU : f (y) = max (0,y), where y is additively shared between S and C
    d. F_sigmoid: approximated sigmoid
    e. F_max : oblivious maxout activation
    f. F_square : oblivious square function
    g. F_trunc : securely scale down the data range without affecting accuracy
7. Iron: Functional Encryption using Intel SGX
8. Global-Scale Secure Multiparty Computation
    a. F_aBIT : distributes autenticated bits to parties
    b. F_pre
    c. F_ashare : authenticated shrae
    d. F_HaAND : Haldf authentivated AND
    e. F_LaAND: leaky AND triple
    f. F_aAND : generating AND triples
    g. F_mpc: mentioned
9. Authenticated Garbling and Efficient Maliciously Secure Two-Party Computation
    a. F_pre, F_abit, F_HaAND, “”, “”
10. Practical UC-Secure Delegatable Credentials with Attributes and Their Application to Blockchain
    a. F_dac: delegatable credentials with attributes
    b. F_crs, F_ca F_smt,
11. FAME: Fast Attribute-based Message Encryption
12. A Framework for Constructing Fast MPC over Arithmetic Circuits with Malicious Adversaries and an Honest-Majority ∗
    a. F_rand generates a sharing [r] of a random value r
13. Fairness in an Unfair World: Fair Multiparty Computation from Public Bulletin Boards
    a. F_att : sgx like behavior
14. Concurrency and Privacy with Payment-Channel Networks
    a. F_pcn : for pcn
    b. Fugor, rayo, etc
    c. F_pcn non blocking
15. Solidus: nConfidential Distributed Ledger Transactions via PVORM
    a. F_init initializer with banks
    b. F_ledger
    c. F_sol : solidus system with banks
16. Malicious-Secure Private Set Intersection via Dual Execution
    a. F_psi one-sided input
    b. F_encode: oblivious encoding

17. Efficient, Constant-Round and Actively Secure MPC: Beyond the Three-Party Case
    a. F_4aot : 4-party attested ot
    b. F_b-4aot : 4-party batch attested ot
18. Fast Private Set Intersection from Homomorphic Encryption
    a. F_psi,
19. DUPLO: Unifying Cut-and-Choose for Garbled Circuits *

CCS 2018

1. PASTA: PASsword-based Threshold Authentication
2. Labeled PSI from Fully Homomorphic Encryption with Malicious Security
    a. F_psi
    b. F_leaky psi
3. Fast Secure Multiparty ECDSA with Practical Distributed Key Generation and Applications to Cryptocurrency Custody *
    a. F_ecdsa, F_com, F_zk, F_com-zk, F_mult, F_checkDH - check h tuple
4. FairSwap: How to Fairly Exchange Digital Goods
    a. F_ledger , F_cfe : coin aided fair exchange
    b. F_jc : judge contact
    c. F_ restricted programmable and observable random oracle
5. General State Channel Networks
    a. F_ch : state channel
    b. F_ledger ??
6. DiSE: Distributed Symmetric-key Encryption
7. An End-to-End System for Large Scale P2P MPC-as-a-Service and Low-Bandwidth MPC for Weak Participants *
8. Ouroboros Genesis: Composable Proof-of-Stake Blockchains with Dynamic Availability
    a. F_init
    b. F_vrf , F_kes : key evolving signature from somewhere else
9. Fast Secure Computation for Small Population over the Internet
    a. F_fair , F_god ??
10. Secure Computation with Differentially Private Access Patterns
    a. F_gas functionality for a single iteration of the GAS model operations
    b. F_dumgen, ?
11.