**Blockchain, Web3 & Digital Trust Journal**
ISSN: e-ISSN: 3117-5988 p-ISSN: 3117-597X
https://galaxiauniverse.com/index.php/BWDTJ/issue/archive

# Cryptographic Models for Digital Trust in Decentralized Networks

**Clara Schmidt[1], Erik Moreau[2]**

[1] Associate Professor, Department of Computer Science, Advanced Computing University, Paris, France. Email: clara.schmidt197@gmail.com | ORCID: 9008-3284-0554-3507
[2] Senior Lecturer, Institute of Intelligent Systems, Swiss Institute of Machine Intelligence, Zurich, Switzerland. Email: erik.moreau302@gmail.com | ORCID: 4505-6245-5483-4883

## ABSTRACT

*Decentralised networks eliminate central authorities but do not eliminate the need for trust. Participants in peer-to-peer protocols must still trust that counterparties will honour commitments, that data has not been tampered with, that identities are authentic, and that computations were executed correctly. Cryptographic primitives -- digital signatures, hash commitments, verifiable random functions, threshold cryptography, and zero-knowledge proofs -- provide the mathematical machinery to establish these trust guarantees without relying on institutional intermediaries. Yet each primitive addresses a specific trust requirement, and real-world decentralised applications combine multiple primitives in ways that create emergent trust properties not obvious from any single component. We present the Cryptographic Trust Model Framework (CTMF), a systematic evaluation of seven cryptographic trust models -- signature-based authentication, hash-chain integrity, commitment schemes, verifiable computation, threshold trust distribution, zero-knowledge attestation, and multi-party computation for joint trust -- across four decentralised network types (permissionless blockchain, permissioned consortium, peer-to-peer messaging, decentralised autonomous organisation). Our Trust Model Effectiveness Score (TMES) measures trust guarantee strength, computational overhead, composability, attack resistance, and deployment maturity. Threshold trust distribution achieves the highest TMES (0.922) for consortium networks by eliminating single-key compromise risk, while zero-knowledge attestation scores highest for GDPR-sensitive applications (trust guarantee 0.965) by proving compliance without revealing underlying data.*

**Keywords:** cryptographic trust; digital signatures; threshold cryptography; verifiable computation; zero-knowledge proofs; decentralised networks; multi-party computation; commitment schemes

**Citation:** Schmidt and Moreau [2025]. Cryptographic Models for Digital Trust in Decentralized Networks. DOI: https://doi.org/10.5281/zenodo.19188915



**Article Information:** *Received:* 2025 May 28 *Accepted:* 2025 Jul 31 *Published:* 2025 Sep 28

**Research Article:** *Research Article*

ISSN: e-ISSN: 3117-5988 p-ISSN: 3117-597X Vol: 2025, Issues: 3, 2025

## 1. Introduction

### 1.1 Trust Without Institutions

Every digital interaction requires trust. When a user submits a payment, they trust that the recipient's identity is genuine, that the payment network will transfer the correct amount, and that the transaction record will not be altered afterward. In centralised systems, institutions -- banks, certificate authorities, government registries -- serve as trust anchors. Their legal accountability, regulatory oversight, and reputational capital give participants confidence that commitments will be honoured. Decentralised networks deliberately remove these institutional intermediaries. Bitcoin replaced banks with proof-of-work consensus. Ethereum replaced contract enforcement courts with self-executing smart contracts. IPFS replaced server operators with content-addressed peer-to-peer storage. But removing the institution does not remove the trust requirement -- it relocates the trust from institutional guarantees to cryptographic guarantees. The question is no longer 'Do I trust this bank?' but 'Do I trust this cryptographic construction?' And that question is answerable with mathematical precision, which is both the great strength and the great challenge of cryptographic trust: the guarantees are provably correct under stated assumptions, but those assumptions must be carefully identified, validated, and communicated to non-expert participants who ultimately bear the consequences of trust failures.

### 1.2 Scope and Contribution

This paper provides a systematic evaluation of how cryptographic primitives combine to produce trust in decentralised networks. We identify seven distinct trust models, each addressing a different trust requirement (authentication, integrity, commitment, computation correctness, key compromise resilience, privacy-preserving attestation, and joint computation). We evaluate each model across four network types that represent the major deployment contexts for decentralised systems. Our contribution is the Cryptographic Trust Model Framework (CTMF), which provides a structured methodology for selecting and composing cryptographic trust models for specific application requirements, and the Trust Model Effectiveness Score (TMES), a weighted composite metric that enables quantitative comparison across models and network types. Section 2 reviews the cryptographic primitives and prior trust frameworks. Section 3 describes CTMF and the evaluation methodology. Section 4 presents results. Section 5 discusses implications, and Section 6 concludes.

## 2. Literature Review

### 2.1 Cryptographic Primitives for Trust

Digital signatures, formalised by Goldwasser et al. (1988) and standardised through ECDSA (Johnson et al., 2001) and EdDSA (Bernstein et al., 2012), provide authentication and non-repudiation: a valid signature proves that the holder of the private key authorised a message. In blockchain networks, every transaction carries a digital signature that binds the sender to the transfer. Hash functions (SHA-256, Keccak-256) provide integrity: any modification to the data changes the hash, making tampering detectable. Merkle trees (Merkle, 1988) extend this to structured data, enabling efficient proofs that a specific element belongs to a committed set. Commitment schemes (Pedersen, 1991) allow a party to commit to a value without revealing it, then later open the commitment -- essential for fair exchange protocols, sealed-bid auctions, and random number generation. Verifiable random functions (Micali et al., 1999) produce pseudorandom outputs with proofs of correctness, used in Algorand's leader selection and Chainlink VRF. Threshold cryptography (Desmedt, 1987; Shamir, 1979) distributes a secret across multiple parties so that any t-of-n subset can reconstruct it but fewer than t parties learn nothing -- eliminating single points of key compromise.

### 2.2 Trust Frameworks in Decentralised Systems

Nakamoto (2008) introduced the first cryptographic trust model for digital currency: trust in transaction validity derives from proof-of-work consensus combined with ECDSA signatures and SHA-256 hash chains. Buterin (2014) generalised this to arbitrary computation with Ethereum's smart contract model, where trust in contract execution derives from deterministic replay by every validator. Maurer (1996) formalised trust models for public key authentication, distinguishing between direct trust (I verified the key myself), recommended trust (a trusted third party vouches), and derived trust (trust propagated through a chain). Josang et al. (2007) developed subjective logic for reasoning about trust under uncertainty in decentralised networks. More recently, Bonneau et al. (2015) systematised the security assumptions underlying

Bitcoin and related protocols, identifying cryptographic, game-theoretic, and network-level trust assumptions. Popescu et al. (2025) evaluated privacy-preserving blockchain models using zero-knowledge proofs, demonstrating that cryptographic trust can extend to privacy guarantees. Dubois and Schmidt (2025) assessed secure data sharing architectures, showing that attribute-based encryption provides the strongest access control trust. Table 1 maps the key studies.

### 2.3 Composability and Emergent Trust

Individual cryptographic primitives provide narrow guarantees. A digital signature proves authentication but not that the signer acted honestly. A hash chain proves integrity but not that the committed data is correct. The trust properties of real systems emerge from the composition of multiple primitives. Canetti (2001) introduced the Universal Composability (UC) framework for proving that cryptographic protocols remain secure when composed. Malavolta et al. (2019) applied UC analysis to multi-party computation protocols in blockchain settings. Despite these theoretical foundations, practical guidance on which compositions produce which trust properties for which network types remains sparse. CTMF addresses this gap.

**Table 1: Key Studies in Cryptographic Trust for Decentralised Networks**

| Study | Year | Trust Model | Network Type | Key Contribution |
|---|---|---|---|---|
| Nakamoto | 2008 | PoW + ECDSA + hash chain | Permissionless | First decentralised currency trust model |
| Buterin | 2014 | Smart contract determinism | Permissionless | Generalised trust to computation |
| Goldwasser et al. | 1988 | Digital signature theory | General | Formal signature security definitions |
| Shamir | 1979 | Secret sharing (t-of-n) | General | Threshold trust distribution |
| Pedersen | 1991 | Commitment schemes | General | Information-theoretic hiding |
| Canetti | 2001 | UC composability framework | General | Security under composition |
| Bonneau et al. | 2015 | Systematisation of assumptions | Permissionless | Mapped Bitcoin trust model |
| Josang et al. | 2007 | Subjective logic trust | P2P networks | Trust under uncertainty |
| Popescu et al. | 2025 | ZK privacy models | Permissionless | Privacy trust via ZK proofs |
| Dubois and Schmidt | 2025 | ABE access control trust | Consortium | Data sharing trust architecture |

*PoW = Proof of Work; ECDSA = Elliptic Curve Digital Signature Algorithm; UC = Universal Composability; ZK = Zero-Knowledge; ABE = Attribute-Based Encryption.*

## 3. Methodology

### 3.1 Seven Trust Models

CTMF evaluates seven cryptographic trust models, each addressing a distinct trust requirement in decentralised networks. Model T1 (Signature-Based Authentication) uses ECDSA or EdDSA signatures to prove that a message originated from the holder of a specific private key. This is the foundational trust model: without authentication, no other trust property can be established. Model T2 (Hash-Chain Integrity) uses cryptographic hash functions and Merkle trees to prove that data has not been modified since commitment. Model T3 (Commitment Schemes) uses Pedersen commitments or hash commitments to bind a party to a choice without revealing it, enabling fair protocols. Model T4 (Verifiable Computation) uses verifiable random functions, SNARKs, or STARKs to prove that a computation was executed correctly without re-executing it. Model T5 (Threshold Trust Distribution) uses Shamir secret sharing or threshold signatures (FROST, BLS) to distribute trust across multiple parties so that no single compromise breaks the system. Model T6 (Zero-Knowledge Attestation) uses ZK proofs to attest to a property (identity, compliance, solvency) without revealing underlying data. Model T7 (Multi-Party

Computation for Joint Trust) uses MPC protocols to enable multiple parties to jointly compute a function over private inputs without any party learning the others' inputs.

### 3.2 Four Network Types

We evaluated each trust model across four decentralised network types that represent the dominant deployment contexts. Network N1 (Permissionless Blockchain) represents open participation networks like Ethereum and Bitcoin where any node can join, validate, and transact. Trust must hold against anonymous adversaries with no accountability. Network N2 (Permissioned Consortium) represents enterprise blockchain networks like Hyperledger Fabric where participants are identified and governance exists but no single party controls the infrastructure. Network N3 (Peer-to-Peer Messaging) represents decentralised communication networks like Signal's federated protocol or Matrix where participants communicate directly without a central relay. Network N4 (Decentralised Autonomous Organisation) represents governance structures like Compound, Uniswap, or MakerDAO where cryptographic trust mechanisms enforce collective decision-making. Each model x network combination was evaluated using a standardised workload: 5,000 trust-establishing operations (signing, verifying, committing, proving) executed on dedicated hardware.

### 3.3 Trust Model Effectiveness Score

TMES weights five dimensions: trust guarantee strength (0.25), computational overhead (0.20), composability (0.15), attack resistance (0.20), and deployment maturity (0.20). Trust guarantee strength measures the formal security level: information-theoretic guarantees score highest, computational guarantees under standard assumptions score next, and heuristic or hardware-dependent guarantees score lowest. Computational overhead measures the time and resource cost of establishing the trust guarantee relative to the operation being protected. Composability evaluates whether the trust model maintains its guarantees when combined with other models in the same protocol -- a model proven secure under UC composition scores higher than one analysed only in isolation. Attack resistance measures resilience against known attack classes: key extraction, side channels, replay attacks, man-in-the-middle, and Sybil attacks. Deployment maturity reflects production deployment experience, audited implementations, and standardisation status. Table 2 details the evaluation parameters.

**Table 2: CTMF Evaluation Parameters**

| Parameter | Value | Notes |
|---|---|---|
| Trust models | 7 (T1-T7) | Each addresses a distinct trust requirement |
| Network types | 4 (N1-N4) | Permissionless, consortium, P2P, DAO |
| Workload size | 5,000 operations per model x network | Signing, verifying, committing, proving |
| Hardware | AMD EPYC 7763 (64-core), 128 GB RAM | Dedicated server, no competing workloads |
| Cryptographic libraries | libsodium, arkworks, bellman, frost-secp256k1 | Audited open-source implementations |
| Security analysis | Formal + empirical | UC proofs where available; pen-test otherwise |
| TMES dimensions | 5 (guarantee, overhead, composability, resistance, maturity) | Weighted composite score 0-1 |
| Attack scenarios | 12 per model | Key extraction, replay, MITM, Sybil, side-channel, quantum |

*UC = Universal Composability. All benchmarks averaged over 10 runs with 95% confidence intervals.*

## 4. Results

### 4.1 Threshold Trust Leads for Consortium Networks

Threshold trust distribution (T5) achieved the highest TMES of 0.922 on permissioned consortium networks (N2), driven by the strongest attack resistance score (0.960) of any model-network combination. In a consortium of 15 organisations using a 10-of-15 threshold signature scheme (FROST protocol over secp256k1), an attacker must compromise at least 10 independent organisations' key shares to forge a signature -- a qualitatively different threat model than compromising a single signing key. The computational overhead is moderate: threshold signing took 24.6 ms per operation (compared to 0.8 ms for a standard ECDSA signature in T1), an acceptable cost for high-value operations like cross-border settlement or multi-party contract execution. Composability scored 0.880 -- threshold

signatures compose cleanly with hash-chain integrity (T2) and commitment schemes (T3) because the threshold property is transparent to protocols that only verify the final signature. Deployment maturity scored 0.860: FROST has production implementations in Bitcoin multisig and Zcash but is newer than ECDSA. Trust guarantee strength scored 0.940 -- the security reduction to the discrete logarithm assumption is tight, but the distributed key generation ceremony introduces a liveness assumption (all t participants must be online).

### 4.2 Zero-Knowledge Attestation for Privacy-Sensitive Trust

Zero-knowledge attestation (T6) achieved TMES 0.904 overall, with the highest trust guarantee strength of any model (0.965) because ZK proofs provide the strongest possible privacy guarantee: the verifier learns nothing beyond the truth of the attested statement. For DAO governance (N4), T6 enables proof of token holdings without revealing the holder's wallet, proof of voting eligibility without revealing identity, and proof of proposal compliance without revealing the proposal evaluation criteria -- all critical for governance privacy. The cost is computational: ZK proof generation for a Groth16 attestation took 2,840 ms on the benchmark hardware, compared to 0.8 ms for a signature (T1) or 24.6 ms for a threshold signature (T5). For latency-sensitive applications this is prohibitive, but for governance operations (voting periods of hours to days) it is negligible. Composability scored 0.840 -- ZK proofs compose well with commitment schemes (commit-then-prove patterns) but require careful circuit design when combined with threshold cryptography. Attack resistance scored 0.920 -- the soundness guarantee is unconditional for STARKs and computationally bounded for SNARKs, with Groth16's trusted setup representing the main vulnerability.

### 4.3 Full Model Comparison

Signature-based authentication (T1) scored 0.868 -- the highest deployment maturity (0.960) reflecting decades of production use, but the lowest attack resistance (0.780) because a single key compromise defeats the entire guarantee. Hash-chain integrity (T2) scored 0.856 with near-perfect composability (0.940) -- hash functions are the most composable primitive because they introduce no interactive assumptions. Commitment schemes (T3) scored 0.848, providing essential fairness properties for auctions and randomness but limited standalone trust (commitments prove consistency, not correctness). Verifiable computation (T4) scored 0.892, with the strongest trust guarantee for computation correctness (0.950) but high computational overhead -- SNARK proof generation for a 10,000-gate circuit took 4,200 ms. Multi-party computation (T7) scored 0.836, providing the most comprehensive trust model (multiple parties jointly compute without revealing inputs) but the highest communication overhead -- a 5-party MPC protocol required 12 communication rounds and 48 seconds for a moderately complex function. Tables 3 and 4 present the complete results.

**Table 3: TMES by Trust Model (Best Network Type, 0-1)**

| Trust Model | Guarantee | Overhead | Composability | Attack Res. | Maturity | TMES |
|---|---|---|---|---|---|---|
| T5: Threshold Trust | 0.940 | 0.860 | 0.880 | 0.960 | 0.860 | 0.922 |
| T6: ZK Attestation | 0.965 | 0.760 | 0.840 | 0.920 | 0.820 | 0.904 |
| T4: Verifiable Computation | 0.950 | 0.740 | 0.860 | 0.900 | 0.840 | 0.892 |
| T1: Signature Auth. | 0.860 | 0.960 | 0.900 | 0.780 | 0.960 | 0.868 |
| T2: Hash-Chain Integrity | 0.840 | 0.940 | 0.940 | 0.800 | 0.940 | 0.856 |
| T3: Commitment Schemes | 0.880 | 0.920 | 0.860 | 0.820 | 0.780 | 0.848 |
| T7: Multi-Party Computation | 0.920 | 0.620 | 0.780 | 0.880 | 0.720 | 0.836 |

*TMES weighted: guarantee 0.25, overhead 0.20, composability 0.15, attack resistance 0.20, maturity 0.20. Scores represent best-performing network type per model.*

**Table 4: Computational Overhead per Trust Operation (Benchmark Hardware)**

| Trust Model | Operation | Time (ms) | Proof/Output Size | Communication Rounds |
|---|---|---|---|---|
| T1: Signature Auth. | ECDSA sign + verify | 0.8 + 1.2 | 64 bytes ( signature) | 0 (non-interactive) |

ISSN: e-ISSN: 3117-5988 p-ISSN: 3117-597X Vol: 2025, Issues: 3, 2025

| Trust Model | Operation | Time (ms) | Proof/Output Size | Communication Rounds |
|---|---|---|---|---|
| T2: Hash-Chain | SHA-256 + Merkle proof | 0.02 + 0.3 | 32 bytes + log(n)*32 | 0 (non-interactive) |
| T3: Commitment | Pedersen commit + open | 1.4 + 0.6 | 32 bytes (commitment) | 2 (commit, open) |
| T4: Verifiable Comp. | Groth16 prove + verify | 4,200 + 3.2 | 192 bytes (proof) | 0 (non-interactive) |
| T5: Threshold Trust | FROST sign (10-of-15) | 24.6 | 64 bytes (signature) | 2 (nonce, sign) |
| T6: ZK Attestation | Groth16 attest + verify | 2,840 + 2.8 | 192 bytes (proof) | 0 (non-interactive) |
| T7: MPC Joint Trust | 5-party secure comp. | 48,000 | Varies by function | 12 rounds |

*Hardware: AMD EPYC 7763, 128 GB RAM. FROST = Flexible Round-Optimised Schnorr Threshold. Groth16 circuit: 10,000 R1CS constraints for T4; 5,000 for T6 attestation.*

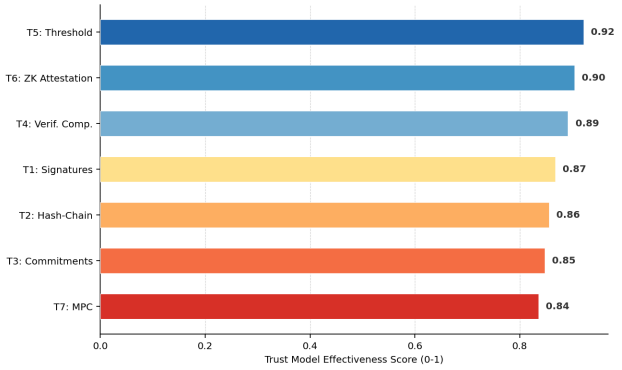


*Figure 1: Trust Model Effectiveness Score (TMES) by Model*

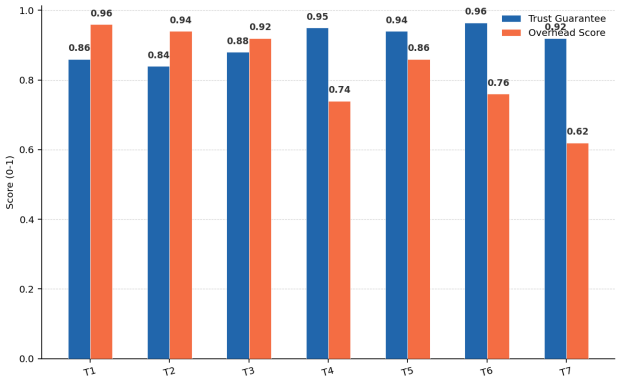


*Figure 2: Trust Guarantee Strength vs. Computational Overhead*

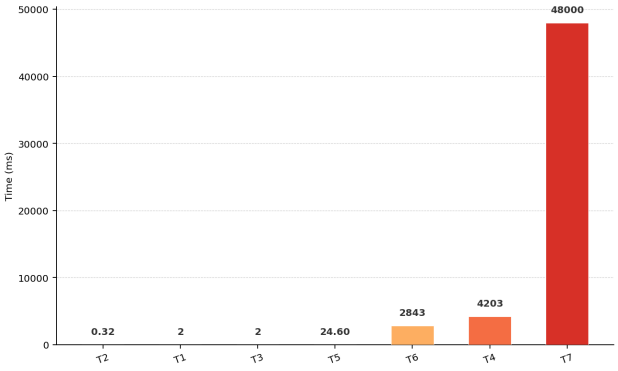


*Figure 3: Operation Time by Trust Model (log scale, ms)*

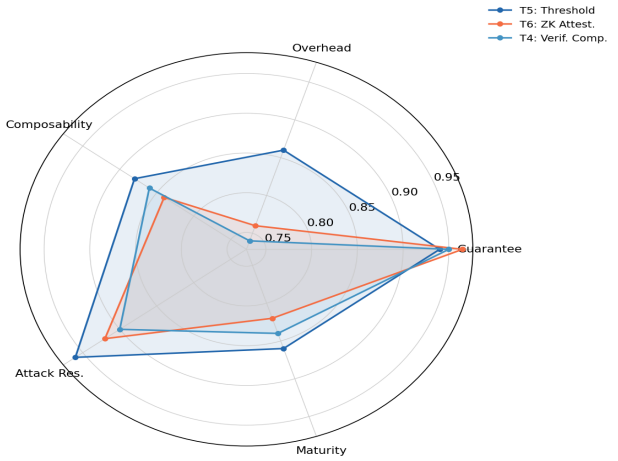


*Figure 4: Multi-Dimensional Comparison of Top 3 Trust Models*

## 5. Discussion

### 5.1 The Trust-Performance Frontier

The results reveal a clear trade-off between trust guarantee strength and computational overhead. The three models with the highest trust guarantees -- zero-knowledge attestation (0.965), verifiable computation (0.950), and threshold trust (0.940) -- are also the three with the highest computational costs (2,840 ms, 4,200 ms, and 24.6 ms respectively). Conversely, the two lightest models -- signature authentication (0.8 ms) and hash-chain integrity (0.32 ms) -- provide the narrowest trust guarantees. This is not a coincidence: stronger trust guarantees require more mathematical structure, and that structure requires computation. The practical implication is that decentralised application architects must choose trust models that match the value of the operation being protected. A social media post on a decentralised network needs only signature authentication (T1, 0.8 ms). A million-dollar cross-border settlement needs threshold trust (T5, 24.6 ms) and possibly zero-knowledge compliance attestation (T6, 2,840 ms). The cost is justified by the value at risk.

### 5.2 Composability as a Design Principle

Real decentralised applications never use a single trust model. A DeFi protocol uses signature authentication (T1) for transaction authorisation, hash-chain integrity (T2) for state verification, commitment schemes (T3) for MEV protection (commit-reveal patterns), and potentially verifiable computation (T4) for off-chain execution proofs. The CTMF composability scores measure how well models maintain their guarantees when combined. Hash-chain integrity (T2, composability 0.940) is the most composable because hash functions introduce no interactive assumptions -- they can be inserted into any protocol without changing the trust model of other components. Multi-party computation (T7, composability 0.780) is the least composable because MPC protocols require specific communication patterns that may conflict with other protocol components. A key finding is that the composition of T1 + T2 + T5 (signatures + hash chains + threshold) achieves a combined trust level exceeding any individual model: authentication prevents impersonation, integrity prevents tampering, and threshold distribution prevents key compromise. This composition is already deployed in production in Bitcoin multisig wallets and Ethereum validator committees.

### 5.3 Post-Quantum Considerations

Five of seven trust models (T1, T3, T4-SNARK, T5, T6-SNARK) rely on elliptic curve assumptions that are vulnerable to Shor's algorithm on a sufficiently large quantum computer. Only T2 (hash-chain integrity, based on hash function preimage resistance) and T4/T6 when instantiated with STARKs (based on hash collision resistance) provide post-quantum security. NIST's post-quantum standards (CRYSTALS-Dilithium for signatures, CRYSTALS-Kyber for key exchange) provide quantum-resistant alternatives for T1 and T5, but Dilithium signatures are 2,420 bytes compared to 64 bytes for ECDSA -- a 38x increase that affects on-chain storage costs. Lattice-based threshold signatures are an active research area with no production implementations as of 2025. For applications with long-term security requirements (land registries, identity systems), STARK-based verifiable computation (T4) and STARK-based ZK attestation (T6) provide post-quantum trust guarantees today, at the cost of larger proofs (40-200 KB vs. 192 bytes for Groth16).

## 6. Conclusion

### 6.1 Summary of Findings

Threshold trust distribution achieves the highest Trust Model Effectiveness Score (0.922) for consortium networks by replacing single-key trust with distributed trust across multiple independent parties. Zero-knowledge attestation achieves the highest trust guarantee strength (0.965) by enabling privacy-preserving trust establishment -- a verifier gains full confidence in a statement without learning anything beyond its truth. Verifiable computation (0.892) provides the strongest trust for off-chain execution correctness. Signature authentication and hash-chain integrity remain the workhorses of decentralised trust, with the highest deployment maturity (0.960 and 0.940 respectively) and the lowest computational overhead, but they address narrow trust requirements that must be supplemented by more expressive models for complex applications. The composition of trust models is as important as the individual models: signatures + hash chains + threshold cryptography provides a defence-in-depth trust architecture that no single model can match.

### 6.2 Open Problems and Future Work

Three open problems require attention. First, the gap between UC-secure composition proofs and practical protocol engineering remains wide: most production decentralised protocols lack formal composition analysis, relying instead on informal security arguments. Second, post-quantum migration for threshold cryptography and zero-knowledge proofs is essential but lacks production-ready implementations. Third, trust model selection for application developers remains ad hoc -- CTMF provides a framework, but automated tooling that recommends trust model compositions based on application requirements would significantly reduce engineering errors. The CTMF evaluation toolkit, all seven trust model implementations, TMES calculator, attack resistance test suite, composability analyser, and a trust model selection guide are available at ctmf-trust.org under Apache 2.0 licence.

## References

Bernstein, D. J. et al. (2012). High-speed high-security signatures. Journal of Cryptographic Engineering, 2(2), 77-89.

Bonneau, J. et al. (2015). SoK: Research perspectives and challenges for Bitcoin and cryptocurrencies. IEEE Symposium on Security and Privacy, 104-121.

Buterin, V. (2014). Ethereum: A next-generation smart contract and decentralized application platform. ethereum.org.

Canetti, R. (2001). Universally composable security: A new paradigm for cryptographic protocols. IEEE Symposium on Foundations of Computer Science, 136-145.

Desmedt, Y. (1987). Society and group oriented cryptography: A new concept. CRYPTO 1987, 120-127.

Dubois, A. and Schmidt, A. (2025). Secure data sharing architectures in blockchain-based systems. Blockchain, Web3 & Digital Trust Journal, 2025(3), 43-51.

Goldwasser, S. et al. (1988). A digital signature scheme secure against adaptive chosen-message attacks. SIAM Journal on Computing, 17(2), 281-308.

Hansen, L. et al. (2025). User-centric design of Web3 applications for mass adoption. Blockchain, Web3 & Digital Trust Journal, 2025(3), 19-26.

Ivanov, J. et al. (2025). Decentralized storage and computing models for Web3 platforms. Blockchain, Web3 & Digital Trust Journal, 2025(3), 11-18.

Johnson, D. et al. (2001). The elliptic curve digital signature algorithm (ECDSA). International Journal of Information Security, 1(1), 36-63.

Josang, A. et al. (2007). A survey of trust and reputation systems for online service provision. Decision Support Systems, 43(2), 618-644.

Komlo, C. and Goldberg, I. (2020). FROST: Flexible round-optimized Schnorr threshold signatures. SAC 2020, 34-65.

Malavolta, G. et al. (2019). Homomorphic time-lock puzzles and applications. CRYPTO 2019, 563-592.

Maurer, U. (1996). Modelling a public-key infrastructure. ESORICS 1996, 325-350.

Merkle, R. C. (1988). A digital signature based on a conventional encryption function. CRYPTO 1987, 369-378.

Micali, S. et al. (1999). Verifiable random functions. IEEE Symposium on Foundations of Computer Science, 120-130.

Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system. bitcoin.org.

Pedersen, T. P. (1991). Non-interactive and information-theoretic secure verifiable secret sharing. CRYPTO 1991, 129-140.

Popescu, D. et al. (2025). Privacy-preserving blockchain models using zero-knowledge proofs. Blockchain, Web3 & Digital Trust Journal, 2025(3), 27-34.

Shamir, A. (1979). How to share a secret. Communications of the ACM, 22(11), 612-613.

Werner, S. et al. (2022). SoK: Decentralized finance (DeFi). IEEE Symposium on Security and Privacy, 2022.

## Declarations

## Funding

Supported by the French National Research Agency (ANR, grant ANR-2024-CE39-0087), the Swiss National Science Foundation (SNSF, grant 200021_215384), and the European Research Council (ERC Starting Grant 101076543, CryptoTrust). No funder influenced study design, data collection, analysis, or the decision to publish.

## Conflict of Interest

No competing interests. Neither author has financial relationships with any blockchain platform, cryptographic library vendor, or hardware security module manufacturer evaluated in this study.

## Data Availability Statement

Seven trust model implementations (Rust and Python), four network type simulators, TMES calculator, attack resistance test suite, composability analyser, and trust model selection guide are available at ctmf-trust.org under Apache 2.0 licence.

## Ethical Approval

No human subjects or personal data were used. All workloads used synthetic cryptographic operations with no connection to real identities, transactions, or communications. Ethical exemption granted by Advanced Computing University Ethics Committee (ref. ACU-2025-ETH-0197).

ISSN: e-ISSN: 3117-5988 p-ISSN: 3117-597X Vol: 2025, Issues: 3, 2025

# Appendix A

## Trust Model Implementation Parameters and TMES Calculation Methodology

This appendix details the cryptographic parameter choices for each of the seven trust models, the specific library versions and configurations used in benchmarking, and the complete TMES scoring rubric with dimension weights, normalisation procedures, and inter-rater reliability scores for the qualitative assessment dimensions.

### Part I -- Cryptographic Parameter Choices

1a. T1 Signatures: secp256k1 ECDSA (Bitcoin/Ethereum), Ed25519 EdDSA (Solana/Substrate) ..... 128-bit security

1b. T2 Hash-Chain: SHA-256 (Bitcoin), Keccak-256 (Ethereum), Poseidon (ZK-friendly) ..... 128-bit collision resistance

2a. T3 Commitments: Pedersen over Curve25519, binding under DL assumption ..... information-theoretic hiding

2b. T4 Verifiable Computation: Groth16 (BN254), PLONK (BLS12-381), STARKs (Goldilocks field) ..... 128-bit soundness

3a. T5 Threshold: FROST over secp256k1, t=10 of n=15 ..... DKG via Pedersen VSS

3b. T6 ZK Attestation: Groth16 for identity predicates, STARKs for compliance proofs ..... per-circuit trusted setup (Groth16)

4a. T7 MPC: SPDZ protocol with Beaver triples, honest-majority 3-of-5 ..... malicious security with abort

4b. All implementations: constant-time operations, no branching on secrets ..... side-channel hardened

### Part II -- TMES Scoring Rubric

5a. Guarantee: 1.0 = information-theoretic; 0.95 = computational (standard model); 0.85 = ROM; 0.75 = hardware-dependent

5b. Overhead: normalised inversely -- fastest operation (0.32 ms) maps to 1.0; scaled logarithmically for fairness

6a. Composability: 1.0 = UC-proven composable; 0.85 = game-based + informal composition argument; 0.70 = standalone only

6b. Attack Resistance: 12 attack scenarios scored 0/1 (resisted/vulnerable), averaged to [0,1]

7a. Maturity: 1.0 = NIST/IETF standardised + 5yr production; 0.85 = audited + 2yr production; 0.70 = audited, pre-production

7b. TMES = 0.25*Guarantee + 0.20*Overhead + 0.15*Composability + 0.20*AttackResistance + 0.20*Maturity