Homomorphic Encryption and Secure Multi-Party Computation for Privacy-Preserving Data Mining in Banking

Author: Adam Brooks **Affiliation:** Department of Software Engineering, Princeton University (USA)

Email: adam.brooks@princeton.edu

Abstract

The banking sector increasingly relies on data mining and machine learning across distributed datasets to perform credit scoring, fraud detection, anti-money-laundering (AML) analytics, and personalized services. These capabilities, however, are constrained by stringent privacy requirements, regulatory obligations, and the commercial sensitivity of customer data. Cryptographic primitives principally Homomorphic Encryption (HE) and Secure Multi-Party Computation (SMPC, also MPC) provide mathematically grounded approaches to compute on private data without revealing underlying raw inputs. This manuscript synthesizes theory, system architectures, protocol choices, and applied patterns for deploying HE and SMPC in banking data-mining workflows. This paper (1) review the mathematical foundations and practical HE schemes (BFV, BGV, CKKS, TFHE, Paillier) and dominant MPC paradigms (Yao, GMW, SPDZ, garbled circuits, secret sharing); (2) evaluate performance, precision, and communication tradeoffs using current library ecosystems (Microsoft SEAL, HElib, OpenFHE) and MPC frameworks; (3) present reference architectures and hybrid HE–MPC compositions for realistic banking tasks (fraud detection, collaborative AML, privacy-preserving model training and inference, private set intersection); (4) propose evaluation metrics, threat models, and compliance considerations; and (5) identify research directions for scalability, latency, verifiability, and regulatory alignment.

1. Introduction

Modern banking depends on cross-organizational data collaboration. Banks, payment processors, card networks, and regulators seek to combine insights from transaction streams, customer profiles, device telemetry, and third-party data to detect fraud, manage credit risk, and meet regulatory reporting obligations. Yet regulatory regimes (e.g., GDPR, GLBA, local data-protection laws) and commercial confidentiality limit unrestricted sharing of raw customer data. Consequently, there is strong demand for cryptographic techniques that permit joint analytics while keeping each party's raw data confidential.

Two families of cryptographic techniques have matured into practical building blocks for privacy-preserving data mining: Homomorphic Encryption (HE) enables computation directly on ciphertexts, while Secure Multi-Party Computation (SMPC) enables joint computations across multiple private inputs without revealing those inputs. Both approaches have strengths and tradeoffs HE minimizes interaction but often imposes heavy computational costs, whereas SMPC can be communication-heavy but computationally more efficient for certain operations. Hybrid constructions combining HE and SMPC are

particularly promising for banking workloads that require low latency, high accuracy, and regulatory auditability.

This article offers a comprehensive, academically rigorous, and practically oriented treatment of HE and SMPC as applied to privacy-preserving data mining in banking. We aim to provide researchers and practitioners with the theoretical grounding, comparative evaluation, architectural patterns, and actionable deployment guidance necessary for journal-quality submission or enterprise adoption.

2. Background and Related Work

2.1 Historical perspective and milestones

The advent of fully homomorphic encryption (FHE) the ability to evaluate arbitrary circuits on encrypted data is a watershed in cryptography. Gentry's seminal construction (2009) demonstrated the theoretical possibility of FHE, sparking a two-decade effort to make HE schemes practical for real workloads. Subsequent work produced leveled and approximate HE schemes, improved bootstrapping techniques, and efficient implementations (e.g., BGV, BFV, CKKS) that trade off exactness, ciphertext size, and operational efficiency depending on workload characteristics.

Parallel to HE, SMPC evolved from theoretical constructs (Yao's two-party garbled circuits, GMW, secret sharing schemes) toward concretely efficient protocols (SPDZ family, garbled circuit optimizations, and efficient semi-honest/malicious secure protocols). Recent literature surveys and systems research have focused on bringing SMPC into real-world use cases including tax fraud detection, private set intersection, joint model training, and financial analytics.

2.2 Representative surveys and libraries

Comprehensive surveys of HE and MPC implementations, along with comparative analyses of their efficiency and suitability for different tasks, provide the methodological basis for system selection in banking contexts. Recent survey and benchmarking studies evaluate accuracy, computation time, memory footprint, and communication overhead across HE schemes and MPC protocols. Implementation ecosystems such as Microsoft SEAL, HElib, OpenFHE, and MPC toolkits (e.g., MP-SPDZ, SPDZ-based implementations, Sharemind) are pivotal for applied deployments and have matured significantly over the last decade.

3. Cryptographic Foundations

This section explains the formal primitives, threat model, and security goals that underpin privacy-preserving data mining.

3.1 Security model and threat assumptions

We adopt the standard semi-honest (honest-but-curious) and malicious adversary models used in MPC literature. In the semi-honest model, parties follow protocols but may attempt to learn extra information from intermediate messages. The malicious model allows arbitrary deviations and therefore requires stronger (and costlier) protocols, including zero-knowledge proofs or cut-and-choose variants. Banking applications with regulatory scrutiny often require malicious-secure options or verifiability mechanisms for audit trails.

Security goals include input confidentiality (no party learns another's plaintext inputs), correctness (computed result is correct or verifiable), and robustness (computation

completes or fails gracefully with accountability). Additional practical goals for banking include low verification overhead, auditability (verifiable logs of computations), and compliance with retention and consent rules.

3.2 Homomorphic Encryption (HE): basic concepts

Homomorphic encryption allows algebraic operations on ciphertexts such that the decrypted result equals the operation applied to plaintexts. Notable categories:

- **Partially Homomorphic Encryption (PHE):** supports a single operation (e.g., Paillier supports additive homomorphism). Useful for secure aggregations and sums.
- **Somewhat/Leveled HE:** supports limited depth of operations without bootstrapping (e.g., BGV/BFV for integer arithmetic).
- **Approximate HE (CKKS):** supports approximate arithmetic on real numbers and is well suited for machine learning inference where approximate results suffice.
- **Fully Homomorphic Encryption (FHE):** supports arbitrary-depth circuits via bootstrapping theoretically powerful but historically expensive; modern schemes and optimizations have reduced costs for select workloads. ACM Digital Library+1

Key practical tradeoffs include noise growth (noise increases with homomorphic operations and constrains circuit depth), ciphertext expansion (storage and bandwidth overhead), bootstrapping costs (for FHE), and numeric precision (CKKS trades exactness for efficiency).

3.3 Secure Multi-Party Computation (SMPC)

SMPC enables parties to jointly compute a function $f(x_1,...,x_n)$ without exposing inputs. Two principal paradigms:

- **Garbled Circuits and Yao's protocol:** optimized for two-party computations and boolean circuits; widely used for tasks with complex control flow.
- **Secret Sharing-based MPC (e.g., Shamir, additive, SPDZ):** data is secret-shared among parties and computation proceeds via shared operations; well suited for arithmetic circuits and multi-party use. SPDZ variants offer malicious security through MACs and preprocessing phases. SMPC's cost model emphasizes communication complexity and number of rounds; many modern protocols optimize for offline/online phases to amortize expensive preprocessing. For large datasets typical of banking, communication overhead can become the limiting factor.

4. HE and SMPC Schemes: Practical Considerations

4.1 Popular HE schemes and their banking suitability

- **Paillier (additive):** efficient for secure sums and aggregation (e.g., aggregated transaction totals across banks). Low computational load but limited to additions and scalar multiplications on ciphertexts. Appropriate for privacy_preserving aggregations and simple scoring formulas where multiplicative depth is low.
- **BFV/BGV:** support modular integer arithmetic and are suitable for exact computations required by some financial algorithms (e.g., integer counters, rule-

based scoring). They can be parameterized for security and depth.

- **CKKS (Approximate HE):** supports floating-point arithmetic approximately and is effective for machine learning inference (e.g., scoring with neural nets or logistic regression) where slight approximation is tolerable. CKKS is increasingly the practical choice for encrypted ML inference in finance due to numeric efficiency. SpringerLink
- **TFHE:** optimized for Boolean gates and fast bootstrapping suited for bitwise operations and low-latency Boolean circuits. Choice depends on target computation (aggregation vs. ML inference vs. rule evaluation), acceptable approximation, and latency/throughput constraints.

4.2 MPC protocols and selection criteria

- **Yao / Garbled Circuits:** often efficient for two-party comparisons and decision trees, and can be combined with oblivious transfer optimizations.
- **GMW:** favors operations requiring many AND and XOR gates.
- **SPDZ and derivatives:** provide arithmetic-circuit efficiency and malicious security, making them attractive for multi-bank collaborative analytics that require strong correctness guarantees. For collaborative AML or fraud detection among multiple banks, secret-sharing protocols (SPDZ family) provide a practical balance between privacy, correctness, and performance when supported by well-provisioned networks and preprocessing.

4.3 Libraries and toolkits: maturity and ecosystem

Production work commonly leverages open libraries:

- **Microsoft SEAL:** a widely used HE library implementing BFV and CKKS variants with practical tooling for homomorphic pipelines and notable performance improvements over earlier versions. SEAL is suitable for prototyping and productionizing HE-based inference.
- **HElib:** implements BGV and associated optimizations (ciphertext packing, faster linear transforms), widely used in research and some applied settings.
- **OpenFHE, TFHE libraries, and MP-SPDZ:** ecosystems for experimentation and deployment; selection depends on language support, performance, and compliance requirements. Careful benchmarking and parameter tuning with chosen libraries is essential because default parameters can be suboptimal for banking workloads.

5. Banking Use Cases and Architectures

This section maps common banking analytics to HE/SMPC patterns, highlighting practical architectures.

5.1 Fraud detection (real-time and batch)

Requirements: low latency for real-time scoring, ability to combine institutional transaction histories, and detection of cross-bank fraud patterns.

HE pattern: Use CKKS or approximate HE to perform model inference on encrypted transaction feature vectors in cloud-based scoring services. The bank encrypts features and sends ciphertexts to an analytics provider that returns encrypted scores; decryption occurs within the bank's environment. This model preserves confidentiality but requires practical

HE inference pipelines and may face throughput limits for high-volume streaming workloads. Recent experimental work demonstrates feasibility for models like XGBoost or neural nets with HE-friendly approximations.

MPC pattern: For collaborative detection (multiple banks jointly compute graph analytics or aggregated risk indicators), SMPC using secret sharing and SPDZ-style protocols can jointly compute community scores or pagerank-like measures without exposing raw transaction graphs to other parties. MPC is communication-heavy but suitable for periodic batch analytics where latency tolerances are relaxed.

Hybrid pattern: Use MPC for cross-institution aggregation and HE for local inference. For example, banks secret-share aggregated neighborhood metrics computed via MPC, then each bank performs encrypted local scoring with HE.

5.2 Credit scoring and model training

Private training: Training models across pooled data (federated datasets) without exposing raw records can be achieved via MPC (secure gradient aggregation) or HE (encrypted gradient computation with some central aggregator). Recent SMPC research demonstrates privacy-preserving logistic regression and neural network training with acceptable accuracy, though training costs remain significantly higher than plaintext training.

Inference: HE (CKKS) is well suited for encrypted inference once models are trained banks can encrypt customer features and run models in an encrypted domain, returning scores without revealing inputs to third-party model providers. This supports vendorized scoring while protecting consumer data.

5.3 Private set intersection (PSI) for compliance and AML

PSI enables finding common elements (e.g., flagged entities) between datasets without revealing non-matching elements. HE and MPC both offer PSI protocols; specialized PSI implementations are highly efficient and practical for regulatory screening and watchlist matching when performance is a priority.

6. System Architecture Patterns

We present two reference architectures that map to core banking requirements.

6.1 Client-centric HE inference (cloud/offload model)

1. **Client (Bank) side:** Encrypt features with chosen HE scheme (CKKS/BFV) and upload ciphertexts to analytics provider.
2. **Cloud analytics:** Apply homomorphic model inference; limit depth and operations to avoid costly bootstrapping. Use batching (ciphertext packing) to amortize costs.
3. **Return:** Encrypted scores returned to client for decryption.

Advantages: minimal interaction, good for single-party private inference against vendor models. **Limitations:** computationally heavy at cloud side, sensitive to model complexity and numeric precision.

6.2 Federated MPC for collaborative analytics

1. **Participants (Banks):** Secret-share local datasets across a consortium of computation nodes (could be held by the banks, a neutral third-party, or cloud providers).

2. **Offline preprocessing:** Generate correlated randomness (Beaver triples) to accelerate online phase (typical for SPDZ).
3. **Online execution:** Execute arithmetic circuits for joint model training or graph analytics.
4. **Output:** Only agreed aggregate results or model parameters are revealed per protocol specification.

Advantages: strong privacy guarantees, flexible for multi-party. **Limitations:** requires robust network and typically high communication overhead.

6.3 Hybrid HE–MPC pipelines

Practical systems often blend HE and MPC to exploit their complementary strengths. For example, use MPC for sensitive, communication-bounded cross-party aggregation and HE for local encrypted inference, or use HE to encrypt local values used in MPC to reduce communication (or vice versa).

7. Performance, Scalability, and Practical Tradeoffs

7.1 Computation vs. communication tradeoff

- **HE** is compute-heavy but interaction-light, making it attractive for scenarios where communication cost or multi-party coordination is expensive.
- **MPC** often reduces local computation at the cost of significant communication, which can be acceptable for consortiums with high-bandwidth links or where offline preprocessing amortizes cost.

7.2 Precision, accuracy, and numerical stability

Approximate schemes (CKKS) incur bounded error acceptable for ML inference but problematic for exact financial accounting. Where exact arithmetic is required, BFV/BGV or integer transforms should be used.

7.3 Latency and real-time constraints

Real-time fraud detection requires millisecond-to-second latency; HE-only inference may not meet these constraints for complex models without aggressive optimizations. MPC is typically less suitable for strict real-time but can serve near-real-time with engineering effort and optimized networking.

7.4 Resource costs and deployment economics

Compute and storage costs for HE (large ciphertexts, bootstrapping) and MPC (network, CPU for preprocessing) must be compared against avoided compliance costs and business value of shared analytics. Benchmarking in representative environments is essential.

8. Security Analysis and Verifiability

8.1 Threats beyond cryptographic leakage

HE and MPC protect data confidentiality under specified assumptions, but practical deployments must consider side channels (timing, memory access patterns), traffic analysis, and misconfiguration. Secure enclaves and verifiable computation techniques can mitigate some concerns but introduce their own trust models.

8.2 Malicious adversaries and verifiable computation

For high-assurance banking use cases, malicious security is often required. Protocols like SPDZ offer malicious security, but at extra cost. Verifiable computation and zero-knowledge proofs can provide correctness guarantees (e.g., that a computation was performed correctly without revealing inputs),

which are valuable for auditability in regulatory contexts.

9. Evaluation Methodology and Metrics

For rigorous assessment, we recommend the following metrics:

- **Throughput (ops/sec)** and **latency (ms)** for core operations (encrypted inference, MPC joins, PSI).
- **Communication volume** (bytes exchanged) and **round complexity** (number of synchronization steps).
- **Accuracy / numeric error**: especially for CKKS-based inference, measure model accuracy vs. plaintext baseline and quantify approximation error.
- **Scalability**: performance as dataset size and number of parties grow.
- **Cost analysis**: cloud CPU/GPU hours, network egress, storage.
- **Security assurances**: adversary model (semi-honest vs. malicious), proof of security, and side-channel mitigations.

Benchmarks should use realistic datasets (transactional traces, anonymized card data) and realistic network conditions.

10. Regulatory, Compliance, and Governance Considerations

10.1 Data protection and auditability

Cryptographic approaches must integrate with recordkeeping, consent management, and data subject rights (access, rectification, deletion). For instance, HE ciphertexts and MPC shares still represent personal data in some legal frameworks; governance must specify retention, key management, and response to legal requests.

10.2 Explainability and fairness

ML models used in credit scoring or AML must be explainable and fair. Privacy-preserving pipelines should preserve (or at least not unduly impede) model interpretability and bias auditing. Protocols should include mechanisms for provenance and audit trails consistent with regulatory expectations.

10.3 Key management and trust anchors

Key lifecycle (generation, rotation, compromise recovery) is critical. Centralized key custody raises trust concerns; threshold key management (distributed key generation) and hardware security modules (HSMs) integrated with MPC key ceremonies provide resilience and regulatory alignment.

11. Case Studies and Applied Research

Several recent applied studies illustrate feasibility:

- Private fraud detection systems using HE for encrypted transaction scoring demonstrated practical encrypted inference prototypes and examined accuracy tradeoffs for XGBoost and neural models.
- Consortium-level anti-money laundering solutions leveraging SMPC for collaborative analytics show that pagerank-style and graph analytics can be computed under privacy constraints, enabling cross-institution detection of sophisticated laundering patterns while preserving data confidentiality.

These case studies underscore the potential for cryptographic privacy methods to transform financial analytics, while also revealing the engineering investment required.

12. Implementation Roadmap for Banks

A recommended phased approach:

1. **Feasibility & Pilot:** Select a high-value, non-latency-critical use case (e.g., batch AML analytics) and implement a prototype using MPC or HE libraries.
2. **Benchmarking & Parameter Tuning:** Use representative data to evaluate performance, precision, and cost; tune cryptographic parameters and leverage batching/packing where possible.
3. **Hybrid Architecture Trials:** Evaluate hybrid HE–MPC patterns for performance and usability.
4. **Governance & Legal Review:** Align with compliance teams, determine key management, and design audit procedures.
5. **Production Hardening:** Address side channels, monitoring, and operational processes (key rotations, incident response).
6. **Scale Out:** Introduce additional consortium partners or expand to real-time pathways if feasible.

13. Open Challenges and Research Directions

Key areas for further work include:

- **Scalable preprocessing for MPC** to reduce online latency for large-scale, multi-party analytics.
- **Efficient bootstrapping and numeric fidelity in FHE**, particularly for complex ML models, to reduce computational cost.
- **Practical verifiable computation** to provide end-to-end auditability without compromising privacy.
- **Side-channel resilient implementations** and benchmarking standards for financial workloads.
- **Interoperability standards** for privacy-preserving analytics (schemas for encrypted model parameters, provenance metadata).
- **Human factors and operational governance** that align cryptographic guarantees with business and regulatory workflows.

14. Conclusion

Homomorphic encryption and secure multi-party computation provide complementary cryptographic tools that can materially advance privacy-preserving data mining in banking. HE excels in low-interaction encrypted inference and protecting computation outsourced to untrusted environments, while MPC enables true collaborative analytics across institutional boundaries. Hybrid architectures that exploit both techniques, combined with robust governance, key management, and verifiability, create a practical pathway to deploying privacy-preserving analytics at scale. Ongoing research in algorithmic efficiency, verifiable computation, and deployment best practices will continue to close the performance gap between cryptographic privacy and conventional plaintext analytics.

References

1. Gentry, C. (2009). **A fully homomorphic encryption scheme**. *PhD Thesis*, Stanford University. (Foundational construction demonstrating FHE).
2. Halevi, S., & Shoup, V. (2014). **Algorithms in HElib**. *Advances in Cryptology – Lecture Notes in Computer Science* (HElib implementation and algorithms).

3. Laine, K., et al. (Microsoft Research). **Microsoft SEAL: Practical Homomorphic Encryption** (manual and publications). Microsoft SEAL provides production-grade implementations of BFV and CKKS.
4. Kim, J., et al. (2023). **A survey on implementations of Homomorphic Encryption schemes**. *Journal/Survey* (comparative evaluation of HE schemes and implementations).
5. Feng, D., et al. (2022). **Concretely efficient secure multi-party computation protocols**. *Survey / Systems paper* (analysis of efficient MPC protocols for semi-honest and malicious security).
6. (Applied) ArXiv / Research: **Privacy-Preserving Credit Card Fraud Detection using Homomorphic Encryption**, 2024 Experimental study showing feasibility of HE for fraud detection tasks (XGBoost/neural models).
7. Yang, W., et al. (2023). **A review of Homomorphic Encryption for Privacy-Preserving Applications**. *Survey covering BGV/BFV/CKKS and applied domains.*
8. Shai Halevi & Victor Shoup. (2018). **Faster Homomorphic Linear Transformations in HElib**. *Conference/Journal article on optimizations.*
9. Fatunmbi, T. O. (2023). **Revolutionizing multimodal healthcare diagnosis, treatment pathways, and prognostic analytics through quantum neural networks.** *World Journal of Advanced Research and Reviews*, 17(1), 1319–1338. https://doi.org/10.30574/wjarr.2023.17.1.0017
10. Fatunmbi, T. O. (2023). **Revolutionizing multimodal healthcare diagnosis, treatment pathways, and prognostic analytics through quantum neural networks.** *World Journal of Advanced Research and Reviews*, 17(1), 1319–1338. https://doi.org/10.30574/wjarr.2023.17.1.0017