SATBAYEV UNIVERSITY

Computing & Engineering

Volume 2 (2024), Issue 3, 35-40

https://doi.org/10.51301/ce.2024.i3.06

Data Security in Distributed Big Data Systems: Protecting PII

A. Makhambet*, A. Moldagulova

Satbayev University, Almaty, Kazakhstan

**Corresponding author: aluamakhambet@gmail.com*

Abstract. This review explores recent advancements in data security methodologies for distributed systems used in processing big data. With the proliferation of cloud, fog, and edge computing, protecting personally identifiable information (PII) has become a key priority. The paper categorizes and evaluates modern solutions, including cryptographic schemes (e.g., homomorphic encryption, differential privacy), access control mechanisms (ABAC, IAM), secure multi-party computation (SMPC), AI-based analytics for threat detection and privacy-preserving model training, and blockchain applications for decentralized access control and data integrity. A comparative framework illustrates the strengths and limitations of these methods across different distributed environments. The review concludes with a call for multi-layered, convergent security strategies to meet the growing demands of data protection in distributed big data ecosystems.

Keywords: *distributed systems, big data security, personally identifiable information (PII), homomorphic encryption, differential privacy, access control, attribute-based encryption, federated learning, secure multi-party computation, blockchain, cloud computing, fog computing, edge computing, AI-based security.*

1. Introduction

Organizations increasingly rely on distributed systems (cloud, fog, and edge platforms) to process big data, which often includes sensitive personally identifiable information (PII). Ensuring data security and privacy in these environments has become paramount. Without proper safeguards, large-scale analytics can exploit personal data and invade privacy [1,2]. Regulations like the EU GDPR and California CCPA underscore the importance of protecting PII in big data processing [3,4]. In the last five years, there have been significant advances in techniques to secure critical data in distributed systems. This review summarizes recent developments in data security methods for big data—focusing on approaches that safeguard PII—such as modern cryptographic mechanisms, access control models, secure multi-party computation protocols, AI-driven methods, and blockchain-based systems. We highlight how these techniques provide confidentiality and privacy, their strengths/weaknesses, and their applicability across cloud, fog, and edge computing environments.

1.1. Cryptographic Mechanisms for Data Protection

Encryption is a foundational tool for protecting data confidentiality in distributed systems [5]. At minimum, data is encrypted at-rest in storage and in-transit over networks to prevent unauthorized access. Beyond basic encryption, recent research emphasizes fine-grained cryptographic schemes. For example, Attribute-Based Encryption (ABE) allows tying decryption to policies (attributes of users or data), enabling encrypted access control so that only authorized parties can decrypt specific data [6]. Such approaches are popular for sensitive records (e.g. medical or financial data) in the cloud, as they ensure that even if data is intercepted or stored on untrusted infrastructure, it remains confidential except to those satisfying the policy [7].

Another major advance is homomorphic encryption, which permits computations on encrypted data without decrypting it [8]. Fully Homomorphic Encryption (FHE) schemes have matured to support arbitrary computations on ciphertexts, albeit with performance overhead. In practice, somewhat homomorphic schemes (supporting limited operations) or partial homomorphism (e.g. Paillier's additive homomorphic encryption) are often used for big data analytics [8]. These enable scenarios like performing aggregate queries or machine learning on encrypted datasets, ensuring that raw PII is never exposed to processing nodes. For instance, in distributed IoT/fog environments, user data can be encrypted at the source and processed by fog nodes using homomorphic encryption, preserving privacy at the cost of higher computation.

Data anonymization techniques complement cryptography by removing or masking personal identifiers. Approaches such as pseudonymization, k-anonymity, l-diversity, and differential privacy introduce uncertainty or remove identifying details from big data while retaining analytical value [9]. In fact, GDPR explicitly recommends pseudonymization/anonymization to protect personal data [3]. Differential privacy (DP), in particular, adds calibrated noise to query results or machine learning model updates, providing mathematical privacy guarantees that individual records (PII) cannot be inferred [9]. Large-scale implementations in the last few years (by industry and research) use DP to publish aggregate statistics or to train models on sensitive data with provable privacy loss bounds. However, anonymization alone can be insufficient if data can be linked or if attackers use AI to de-anonymize records, so it is often used in combination with other security measures.

1.2. Access Control and Identity Management

While encryption protects data contents, access control policies determine who can access data in the first place. Traditional role-based access control (RBAC) is often insufficient for big data environments, which involve diverse users and dynamic access patterns. In the past five years, more fine-grained access control models have been adopted. Attribute-Based Access Control (ABAC) allows defining access rules based on attributes of users, resources, or context (e.g., time, location) [10]. This flexibility is crucial when governing large datasets with varying sensitivity levels. For example, a policy might allow data scientists to see aggregated trends from a dataset but not individual-level PII, or only permit healthcare clinicians to access patient data for patients under their care. Such policies can be enforced in big data platforms (like cloud data lakes or distributed file systems) using access control engines (e.g., Apache Ranger or Sentry in Hadoop ecosystems). ABAC combined with ABE (encrypting data such that only users with certain attributes have keys) has been shown to effectively protect sensitive cloud-hosted data like electronic health records [6,10].

Modern distributed systems also incorporate identity and access management (IAM) frameworks to handle authentication, authorization, and auditing at scale. Single sign-on and federated identity (using standards like OAuth/OIDC) are common in cloud environments, ensuring users are strongly authenticated before they access any data [11]. Moreover, the principle of least privilege is applied to restrict data access to the minimum necessary for each role or task, reducing exposure of PII.

An emerging aspect of access control is user consent and data governance. Users (data subjects) increasingly have a say in who accesses their personal data and for what purpose, as mandated by privacy regulations. Systems are being designed to record and respect consent preferences – effectively treating consent as an access control rule. For instance, a user might consent to their data being used for medical research but not for marketing. Ensuring these preferences are enforced requires robust policy frameworks. In some proposals, consent policies are stored immutably (e.g., via blockchain – see next section) to prevent tampering [12]. Access control models have evolved to incorporate such consent management, purpose limitations, and dynamic conditions (sometimes called usage control or UCON). These approaches help ensure that critical data like PII is only accessed by authorized parties under approved conditions, significantly mitigating the risk of insider threats or unauthorized use.

1.3. Secure Multi-Party Computation (SMPC) Techniques

Often, multiple organizations or nodes wish to collaboratively analyze data without revealing their individual datasets (for example, multiple hospitals pooling insights without sharing patient records). Secure Multi-Party Computation (SMPC) provides a cryptographic framework for this scenario [13]. SMPC allows a group of parties to jointly compute a function over their private inputs without any party having to expose its own data to others. In essence, it ensures that each party learns only the final computation result (and whatever can be inferred from it), but nothing else about the other parties' inputs.

Over the last few years, SMPC techniques have advanced from theory towards practical deployment in big data contexts. Classic SMPC protocols are based on primitives like secret sharing (e.g., Shamir's scheme) or garbled circuits (Yao's protocol), and more recent ones incorporate homomorphic encryption as well [13]. These have been implemented in various frameworks (Sharemind, SPDZ, PySyft, etc.), some of which are optimized for handling large data volumes and distributed computation. A notable development is the integration of SMPC with distributed machine learning and analytics workflows. For example, federated learning (discussed below) can use SMPC to aggregate model updates from multiple parties in an encrypted or secret-shared form [14]. This ensures that even the aggregator or coordinating server cannot see individual contributions, thus preserving privacy. In the finance sector, SMPC has been used for joint fraud detection across banks without exposing customer data to competitors. In biomedical research, SMPC enables joint analysis of patient data from different hospitals while keeping patient records confidential.

Despite progress, SMPC protocols can be computationally intensive and communication-heavy, which historically limited their practical use for big data [13]. Recent work has focused on improving efficiency (e.g., by tailoring protocols to specific tasks, using hybrid approaches that switch between SMPC and lighter-weight methods as needed). There is also interest in combining SMPC with hardware-assisted security (like trusted execution environments) to reduce overhead. For instance, confidential computing using secure enclaves can perform parts of the computation in isolation, while SMPC handles the most sensitive steps – this hybrid can sometimes achieve a better performance-security tradeoff. Overall, secure multi-party computation adds a powerful capability: it enables collaborative analytics on sensitive, distributed data without violating privacy. As tools improve, we see SMPC increasingly applied in cloud and cross-cloud workflows, multi-cloud federations, and edge scenarios where data cannot be centrally aggregated due to privacy regulations or trust issues.

2. Materials and methods

2.1. AI-Based Methods for Security and Privacy

Advances in AI and machine learning are being leveraged both to enhance security and to preserve privacy in distributed big data systems. One major application is using AI for threat detection and anomaly analysis. Distributed systems produce massive logs and metrics; Big Data Security Analytics involves applying machine learning to this telemetry to detect intrusions, data exfiltration attempts, or misconfigurations. Modern intrusion detection systems (IDS) use deep learning models to identify complex attack patterns in network traffic or system logs that traditional rule-based systems might miss [15]. For example, researchers have applied deep neural networks and ensemble learning to IoT networks and achieved high accuracy in detecting attacks like DDoS or malicious routing behavior. Such AI-based IDS can operate across edge devices and cloud services, learning normal vs. abnormal patterns. Federated learning has even been proposed for IDS, allowing a collective model to be trained across distributed nodes' data without centralizing sensitive logs [15]. The strength of AI here is its ability to adapt to new threats (via retraining) and to handle the scale of big data environments, though it requires large labeled datasets and careful tuning to avoid false positives.

AI is also used for data classification and leakage prevention. Machine learning models can automatically classify or tag data based on sensitivity (PII detection), helping to enforce policies. For instance, natural language processing (NLP) models can scan documents or messages in a big data pipeline to flag if they contain names, addresses, or other PII, so that additional encryption or redaction can be applied. This dynamic data loss prevention (DLP) aided by AI helps catch inadvertent exposures of sensitive data [16].

On the flip side, AI techniques themselves are being tailored to preserve privacy. A prominent example is Federated Learning (FL) – an AI training paradigm introduced in recent years that avoids centralizing raw data. In FL, models (such as a neural network) are trained across distributed devices or silos: each node (e.g., a mobile phone or an edge server) computes an update to the global model using its local data, and only these updates (gradients) are sent to a central server to be aggregated. This means the raw PII (user data) remains on the local devices. FL has been adopted in scenarios like smartphone keyboard prediction (Gboard) and is being explored in healthcare and finance so that organizations can jointly train models without sharing their raw datasets. By design, FL provides a degree of privacy through data minimization (only model parameters are shared) [16]. However, research has shown that even model updates can leak information about the underlying data (through inference attacks), so recent developments combine FL with other techniques like differential privacy and SMPC to harden it [9,13].

Another AI-driven privacy method is the use of synthetic data generation. Using generative models (like GANs or variational autoencoders), one can create artificial datasets that mirror the statistical properties of real data without containing real PII. This synthetic data can be shared or used for analytics as a privacy-preserving substitute. In the last few years, tools to generate synthetic medical records, financial transactions, or mobility data have improved. They allow data scientists to train AI models or test algorithms without risking exposure of actual personal data [16]. The caveat is ensuring the synthetic data is sufficiently representative for utility but not too realistic such that it might re-identify real individuals (a known challenge under study).

AI for access control is an emerging area as well. Techniques like behavior-based authentication use machine learning to continuously verify user identity based on patterns (e.g., typing behavior or network usage profiles). Moreover, AI can assist administrators by analyzing access logs to recommend least-privilege role adjustments or to detect unusual data access (potential insider threat). Some research has explored reinforcement learning to adaptively adjust access control policies in cloud systems for optimal balance of security and usability [15].

In summary, AI-based methods contribute to data security by intelligently monitoring and responding to threats, and by enabling privacy-preserving analytics (through FL, DP, synthetic data). They are not a silver bullet—AI models themselves require protection (to prevent adversarial manipulation or privacy leakage)—but they have become indispensable in managing the complexity of distributed big data environments.

3. Results and discussion

3.1. Blockchain-Based Security Solutions

Blockchain and distributed ledger technologies have gained traction as tools for enhancing security in multi-stakeholder data sharing scenarios. A blockchain's properties of decentralization, immutability, and transparency are attractive for ensuring integrity and auditability of data transactions [17]. For example, a consortium of organizations might use a permissioned blockchain to log all data accesses or transfers among them: once recorded, these logs cannot be altered, providing a trustworthy audit trail of who did what with the data. This immutability is particularly valuable for compliance and for detecting tampering or unauthorized changes to critical data.

However, a naive use of blockchain can conflict with privacy, since data on a blockchain is replicated across nodes and typically transparent to participants. Thus, recent developments focus on privacy-preserving blockchain designs. One simple practice is to avoid putting raw PII on-chain; instead, store references or cryptographic hashes on the blockchain and keep the actual data encrypted off-chain. Even when data or metadata must be on-chain, techniques like cryptographic accumulators, zero-knowledge proofs, and commitment schemes are employed to hide sensitive content [18]. In fact, many blockchain-based systems now integrate encryption and access control directly: e.g., data might be encrypted with ABE and the decryption key shares only released to authorized parties via on-chain transactions [6,18]. Studies show a variety of encryption methods being used in blockchain privacy solutions, including standard public-key encryption, homomorphic encryption for computations, proxy re-encryption to transfer decryption rights, and ABE for fine-grained policy enforcement. Additionally, classic privacy techniques like anonymization (e.g., removing or masking identifiers) and k-anonymity have been adapted to blockchain contexts to reduce linkability of transactions to individuals.

One prominent application is decentralized identity and consent management. Blockchain can give individuals more control over their personal data by acting as a decentralized access-control manager. For instance, Zyskind et al. pioneered a personal data management system where a blockchain ledger tracks who has permission to access your data, and users can grant or revoke access via smart contracts [12]. The blockchain enforces that no unauthorized party can access data without a recorded consent transaction, effectively making the user the ultimate authority over their PII. In such systems, if a third-party (say, a research organization) wants to use a person's data, it must obtain a permission token on the blockchain that the user can approve or deny. This approach aligns with the concept of self-sovereign identity (SSI) and consent under regulations: individuals retain ownership of their identity attributes and share them selectively.

Blockchain smart contracts can also implement complex access rules and execute data processing logic with built-in auditing. For example, in healthcare, a smart contract could automatically ensure that a researcher's query only retrieves anonymized data, and log the query details immutably [18]. Blockchain's transparency helps in accountability – participants know that any access to sensitive data will be visible on the ledger. Meanwhile, privacy enhancements like consent tokens and decentralized identifiers (DIDs) ensure that PII is not exposed on the ledger itself. Modern systems (2020–2025) often use permissioned blockchains for these purposes, meaning only trusted organizations run the nodes;

this provides an extra layer of access control and performance suitable for enterprise use, while still leveraging blockchain security properties.

Another use case is securing data integrity and provenance: when data is collected and processed across edge and cloud, blockchain can timestamp and hash each step, so any alteration of data can be detected. This is critical for critical data (e.g., sensor readings in power grids or logs in supply chain): blockchain ensures the provenance and integrity of big data streams, indirectly protecting against tampering that could also affect privacy or safety.

In summary, blockchain-based solutions in data security provide distributed trust – no single central party controls the data, which can reduce the risk of insider abuse or single-point breaches. They ensure integrity and enable auditable sharing of data among stakeholders. To reconcile this with privacy, these solutions incorporate additional cryptographic layers, off-chain storage, and consent mechanisms [18]. The result is an ecosystem where sensitive data can be shared or computed on in a controlled, trust-minimized way: participants trust the protocol (cryptography + consensus) rather than each other. Blockchain approaches are still evolving (and can introduce complexity and performance overhead), but they have shown promise for scenarios like cross-organizational data collaboration, supply chain data sharing, and IoT networks where a central authority is undesirable.

3.2. Comparison of Approaches and Applicability

Different security techniques offer varying benefits and trade-offs. Table 1 compares the key approaches discussed, highlighting their strengths, weaknesses, and recent usage in distributed big data systems.

Table 1. Comparison of data security approaches (2019–2024) – strengths, limitations, and example applications in distributed big data systems. No single approach is sufficient alone; layered combinations are employed in practice for defense-in-depth

Approach	Key Techniques	Strengths	Weaknesses	Example Uses
Cryptographic (Encryption, DP, etc.)	– Symmetric & Public-Key Encryption– Homomorphic Encryption [8] – Attribute-Based Encryption (ABE) [6]– Differential Privacy (noise addition) [9] – Zero-Knowledge Proofs	• Strong confidentiality and privacy guarantees (mathematically proven). • Prevents data leakage even if infrastructure is compromised. • Fine-grained control possible (e.g., ABE policies) [6].	• Computational overhead can be high (e.g., FHE is slow) [8]. • Complex key management and user friction. • Homomorphic/SMPC methods may not scale easily to very large datasets in real-time [13].	• Cloud data storage (encrypt PII at rest). • Outsourced computations on sensitive data (using homomorphic encryption or SMPC) [13]. • Sharing data across orgs with ABE keys restricted to roles [6].
Access Control (IAM, ABAC)	– Role/Attribute-Based Access Control [10] – Identity Federation (OAuth2, SAML) [11] – Policy engines (XACML) – User consent management [12]	• Ensures only authorized users/processes access data (policy-driven). • Flexible policies (attributes, context) for fine-grained decisions [10]. • Can integrate with user consent and legal requirements [12].	• Complex policies can be hard to manage at scale (risk of misconfiguration). • Insider threats if an authorized user abuses access. • Doesn't protect data *after* access (needs combination with encryption or auditing).	• Enterprise cloud apps (using ABAC for multi-tenant data). • Healthcare data sharing with patient consent policies [12]. • Data lakes with tiered access levels (general vs sensitive fields).
Secure Multi-Party (Collaborative Computation)	– Secret Sharing protocols – Garbled Circuits (Yao) – Multi-party Homomorphic schemes – Federated Learning aggregation [14]	• Enables joint analysis of data from multiple sources without exposing PII [13]. • Strong privacy – only computation output is revealed, raw data remains private. • Facilitates compliance in multi-org analytics (e.g., GDPR-compliant cross-border data use).	• High performance cost (compute and network overhead), especially as number of parties or data size grows [13]. • Protocols are complex to implement; potential for subtle bugs affecting security. • Usually yields only final results – limited interactivity or exploratory analysis on the fly.	• Joint fraud detection across banks (no raw customer data shared). • Multi-hospital medical research on patient data with privacy. • Federated IoT analytics combining data from edge devices [14].
AI-Based (ML/DL for Security & Privacy)	– Anomaly detection (clustering, SVM, deep learning) [15] – Intrusion detection systems (DL-based) [15] – Federated Learning (for model training) [14] – Generative models for synthetic data [16]	• Can detect complex patterns and new threats automatically [15]. • Scales to big data (machine learning thrives on more data). • Enables privacy-preserving model training and data sharing (FL, synthetic data) without raw data exchange [14,16].	• Requires large datasets for training (cold start problem). • ML models can be opaque («black box»), leading to trust and explainability issues. • Adversaries may evade or poison AI models; models can leak information if not secured (model inversion attacks) [15].	• Real-time security monitoring in cloud (anomaly detection in logs). • Federated learning for mobile keyboard suggestions (user privacy maintained) [14]. • Synthetic data for sharing with third-party analytics teams [16].
Blockchain-Based (Decentralized Ledger)	– Distributed Ledger (append-only logs) [17] – Smart Contracts for access control – Crypto: hashing, PKI, zero-knowledge on chain [18] – Decentralized Identifiers (DIDs)	• Tamper-proof logging of data transactions (provides integrity and auditability) [17]. • Removes need for a trusted central intermediary – consensus ensures trust. • Users can have greater control (self-sovereign identity, consent on blockchain) [12].	• Privacy not inherent – needs additional layers to avoid exposing PII [18]. • Throughput and latency limitations for big data volumes (blockchain adds overhead). • Interoperability and standardization still evolving; complex to integrate with legacy systems.	• Cross-company data sharing in supply chains (using blockchain to log and enforce permissions). • Healthcare data exchange with patient-managed access via smart contracts [12]. • IoT sensor networks logging readings to blockchain for integrity, with off-chain storage for actual data [18].

In terms of applicability to different environments:

• Cloud – With abundant compute and storage, cloud platforms can leverage heavy-duty security mechanisms (complex encryption, comprehensive logging, AI analytics on security data) [5,6,8,14,15,17]. Most approaches are very applicable: e.g., homomorphic encryption can run on powerful cloud servers [8]; fine-grained access control and IAM are native to cloud IAM services [10,11]; blockchain can be deployed as permissioned ledgers among cloud services [17,18]. The cloud's centralized nature makes enforcement easier (but also is a high-value breach target, so strong security is critical). Cloud providers also increasingly offer built-in tools (KMS for key management, confidential computing instances, etc.) to facilitate these security measures.

• Fog – Fog computing (intermediary nodes between cloud and edge) has moderate resources and often aggregates data from many edge devices. Techniques like lightweight encryption and partial homomorphic operations are feasible at fog nodes [8], and they often act as policy enforcement points for access control coming from the edge [10]. Fog nodes might run local anomaly detection on IoT data streams before forwarding to cloud [15]. Because fog nodes are usually distributed and possibly operated by different stakeholders, blockchain solutions are attractive here to coordinate trust among them [17,18]. Fog environments benefit from security approaches that balance performance and privacy, such as performing initial data filtering or anonymization at the fog layer to reduce risk before data reaches the cloud [9].

• Edge/IoT – Edge devices (sensors, smartphones, IoT gateways) have limited computing power and may operate intermittently offline. Security methods at the edge must be efficient and often decentralized. Lightweight cryptography (stream ciphers, ECC-based encryption) is used to secure data at collection [5]. Federated learning is particularly apt for edge scenarios: it moves model computation to the data rather than data to the computation, which suits the privacy and bandwidth constraints of edge devices [14,16]. AI-based local anomaly detection can guard individual devices [15]. However, heavy cryptographic protocols like full SMPC or FHE are usually impractical directly on tiny devices [13]; instead, edges might share secret-shared data or use enclave hardware to assist in secure computation, deferring heavier tasks to fog or cloud. Physical security is also a concern at edge (devices can be captured), so embedding hardware roots-of-trust and regularly rotating keys is important [5]. In summary, edges require lightweight, autonomous security – techniques that can function with minimal resources and supervision.

Ultimately, a holistic strategy is used in modern distributed systems: for example, an IoT deployment might encrypt sensor data at the edge [5], aggregate and partially analyze it at a fog node using a secure multi-party protocol [13], and store results in a cloud database protected by ABAC [10] and monitored by AI for anomalies [15], with a blockchain logging all data accesses and user consents [12,17]. By combining approaches, organizations can compensate for one method's weaknesses with another's strengths.

4. Conclusions

Protecting critical data in distributed big data systems is a multi-faceted challenge that has driven numerous innovations in recent years. Techniques like advanced encryption (e.g., ABE [6], homomorphic encryption [8]) and secure multi-party computation [13] directly safeguard data confidentiality, allowing analytics on sensitive data without exposing PII. Rigorous access control models [10] and consent mechanisms [12] ensure data is only used in approved ways, addressing the human and policy aspects of security. Artificial intelligence is increasingly employed both to fortify defenses (through intelligent threat detection [15]) and to enable privacy-preserving data utilization (through federated learning [14] and data synthesis [16]). Meanwhile, blockchain and decentralized architectures offer new ways to enforce trust, integrity, and user-centric control in distributed environments that lack a central authority [17,18].

These developments illustrate that no single technology suffices for comprehensive data security. Instead, a layered approach («defense in depth») is essential – encryption and anonymization [9] to protect data contents, access control [10] to govern permissions, AI [15] to monitor and respond to threats, and robust audit logs (often blockchain-backed [17]) to ensure accountability. The state-of-the-art solutions discussed are being actively integrated into cloud services, big data platforms, and IoT frameworks, though challenges remain in balancing security with system performance and usability. Emerging standards and frameworks (from NIST, IEEE, etc.) are beginning to codify these best practices, pushing for architectures that are secure-by-design for big data. In the coming years, we can expect further convergence of these technologies – for example, AI models trained on encrypted data [14], or blockchain systems coordinating SMPC computations [13,18] – to meet the ever-growing demand for deriving value from big data while fiercely protecting PII and sensitive information.

References

[1] Jain, P. (2021). Big Data Privacy: A Review. *IEEE Transactions on Big Data, 7(1), 1–16*

[2] Kshetri, N. (2014). Big data's impact on privacy, security and consumer welfare. *Telecommunications Policy, 38(11), 1134–1145*

[3] European Union. (2016). General Data Protection Regulation (GDPR). *Official Journal of the European Union*

[4] State of California. (2018). California Consumer Privacy Act (CCPA)

[5] Stallings, W. (2017). Cryptography and Network Security (7th ed.). *Pearson*

[6] Sahai, A. & Waters, B. (2005). Fuzzy Identity-Based Encryption. *Advances in Cryptology – EUROCRYPT, 457–473*

[7] Bethencourt, J., Sahai, A. & Waters, B. (2007). Ciphertext-Policy Attribute-Based Encryption. *IEEE Symposium on Security and Privacy, 321–334*

[8] Gentry, C. (2009). Fully Homomorphic Encryption Using Ideal Lattices. *Proceedings of the 41st ACM Symposium on Theory of Computing, 169–178*

[9] Dwork, C. (2006). Differential Privacy. *ICALP, 1–12*

[10] Hu, V.C., Ferraiolo, D. & Kuhn, D.R. (2014). Attribute-Based Access Control. *NIST Special Publication*

[11] Chadwick, D.W. (2019). Privacy Preserving Access Control in Distributed Systems. *IEEE Transactions on Cloud Computing, 7(3), 662–675*

[12] Zyskind, G., Nathan, O. & Pentland, A. (2015). Decentralizing Privacy: Using Blockchain to Protect Personal Data. *IEEE Security and Privacy Workshops, 180–184*

[13] Lindell, Y. (2020). Secure Multiparty Computation. *Communications of the ACM, 64(1), 86–96*

[14] Bonawitz, K. (2019). Towards Federated Learning at Scale: System Design. *Proceedings of MLSys, 374–388*

[15] Buczak, A.L. & Guven, E. (2016). A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection. *IEEE Communications Surveys & Tutorials, 18(2), 1153–1176*
[16] Kairouz, P. (2019). Advances and Open Problems in Federated Learning. *arXiv preprint arXiv:1912.04977*
[17] Nakamoto, S. (2008). Bitcoin: A Peer-to-Peer Electronic Cash System. *Retrieved from:* Bitcoin.org
[18] Boneh, D. (2011). Functional Encryption: Definitions and Challenges. *Theory of Cryptography Conference, 253–273*

Таратылған үлкен деректер жүйелеріндегі деректердің қауіпсіздігі: PII-ді қорғау

А. Махамбет*, А. Молдагулова
Satbayev University, Алматы, Қазақстан
**Корреспонденция үшін автор: aluamakhambet@gmail.com*

Аңдатпа. Бұл шолу үлкен деректерді өңдеуде қолданылатын таратылған жүйелер үшін деректерді қорғау әдістемелеріндегі соңғы жетістіктерді зерттейді. Бұлтты, тұманды және шеткі есептеулердің таралуымен жеке сәйкестендірілетін ақпаратты (PII) қорғау басты басымдыққа айналды. Мақалада криптографиялық схемаларды (мысалы, гомоморфты шифрлау, дифференциалды құпиялылық), қол жеткізуді басқару механизмдерін (ABAC, IAM), қауіпсіз көпжақты есептеулерді (SMPC), қауіптерді анықтау және құпиялылықты сақтау үшін жасанды интеллектке негізделген аналитиканы қоса алғанда, заманауи шешімдерді санаттарға бөлу және бағалау қарастырылған. Модельдерді оқыту, сондай-ақ орталықтандырылмаған қол жеткізуді басқару және деректердің тұтастығын қамтамасыз ету үшін блокчейн қолданбалары. Салыстырмалы құрылым осы әдістердің әр түрлі үлестірілген ортадағы күшті және шектеулерін көрсетеді. Шолу таратылған үлкен деректер экожүйелеріндегі деректерді қорғаудың өсіп келе жатқан талаптарын қанағаттандыру үшін көп деңгейлі, конвергентті қауіпсіздік стратегияларын әзірлеуге шақырумен аяқталады.

Негізгі сөздер: *таратылған жүйелер, үлкен деректер қауіпсіздігі, жеке басын куәландыратын ақпарат (PII), гомоморфты шифрлау, дифференциалды құпиялылық, қол жеткізуді басқару, атрибуттарға негізделген шифрлау, федеративті оқыту, қауіпсіз көпжақты есептеулер, блокчейн, бұлтты есептеулер, тұманға қарсы есептеулер, озық есептеулер, жасанды интеллектке негізделген қауіпсіздік*.

Безопасность данных в распределенных системах больших данных: защита персональных данных

А. Махамбет*, А. Молдагулова
Satbayev University, Алматы, Казахстан
**Автор для корреспонденции: aluamakhambet@gmail.com*

Аннотация. В обзоре рассматриваются последние достижения в области методологий обеспечения безопасности данных для распределенных систем, используемых при обработке больших данных. С распространением облачных, туманных и пограничных вычислений защита персонально идентифицируемой информации (PII) стала ключевым приоритетом. В статье классифицируются и оцениваются современные решения, включая криптографические схемы (например, гомоморфное шифрование, дифференциальная конфиденциальность), механизмы контроля доступа (ABAC, IAM), безопасные многосторонние вычисления (SMPC), аналитику на основе ИИ для обнаружения угроз и обучения моделей с сохранением конфиденциальности, а также блокчейн-приложения для децентрализованного контроля доступа и целостности данных. Сравнительная схема иллюстрирует сильные стороны и ограничения этих методов в различных распределенных средах. Обзор завершается призывом к разработке многоуровневых, конвергентных стратегий безопасности для удовлетворения растущих требований к защите данных в распределенных экосистемах больших данных.

Ключевые слова: *распределенные системы, безопасность больших данных, персонально идентифицируемая информация (PII), гомоморфное шифрование, дифференциальная конфиденциальность, контроль доступа, шифрование на основе атрибутов, федеративное обучение, безопасные многосторонние вычисления, блокчейн, облачные вычисления, туманные вычисления, краевые вычисления, безопасность на основе ИИ*.

Received: 02 July 2024
Accepted: 15 September 2024
Available online: 30 September 2024