A SECURE AND SCALABLE PUBLIC AUDITING SCHEME FOR PRIVACY-PRESERVING MEDICAL DATA STORAGE IN THE CLOUD

[1] Mr. P.Kishore, [2] Kondakindi Sai Kumar Reddy,, [3] Kolanu Pravallika, [4] Gangadi Lara, [5] Indrakanti Deepak Reddy

[1]Assiatant Professor in Department of CSE TKR COLLEGE OF ENGINEERING & TECHNOLOGY

[1]ponnamkishore@tkrcet.com

[2,3,4,5]UG Scholars in Department of CSE TKR COLLEGE OF ENGINEERING & TECHNOLOGY

[2]saikumarreddykondakindi@gmail.com , [3] gangadilara@gmail.com , [4] kolanupravallika23@gmail.com , [5] indrakantideepakreddy10@gmail.com

Abstract

The growing reliance on cloud computing has made it an essential platform for storing and managing large-scale data, particularly in sensitive domains such as healthcare. While cloud storage offers flexibility and cost advantages, it also raises serious concerns about data security, privacy, and integrity. Medical records, in particular, contain highly confidential information that must be protected not only from unauthorized access but also during verification processes. Existing auditing methods often fail to balance efficiency with privacy, as they either expose sensitive data to auditors or introduce significant computational overhead secure and scalable public auditing scheme is presented to address these challenges in cloud-based medical data storage. The proposed approach leverages an identity-based cryptographic framework to simplify key management and remove the dependency on traditional certificate-based systems. By integrating privacy-preserving techniques such as data blinding and sanitization, the scheme ensures that sensitive patient information remains concealed even during the auditing process. At the same time, it enables a third-party auditor to verify data integrity without direct access to the original content or the identity of the data owner.The system is further designed to handle real-world requirements by supporting batch auditing and dynamic data operations, allowing multiple audit tasks to be processed efficiently while maintaining flexibility for data updates. Experimental evaluation and analysis indicate that the approach provides strong security guarantees with reduced computational cost, making it suitable for practical deployment. Overall, this work offers a balanced solution that ensures confidentiality, integrity, and efficiency in cloud-based medical data management.

Keywords

Cloud storage, Medical data security, Privacy preservation, Public auditing, Identity-based cryptography, Data integrity, Third-party auditor, Secure data sharing, Batch verification, Cloud security

I INTRODUCTION

Cloud computing has gradually shifted from a supporting technology to a core infrastructure for storing and managing digital information. Organizations increasingly rely on remote servers to handle large datasets because it reduces operational costs and simplifies maintenance. Instead of maintaining dedicated hardware, users can access shared resources whenever required. The concept is formally described as an on-demand service model that provides flexible access to computing resources over the internet [1].

At the same time, moving data to the cloud introduces a level of uncertainty that did not exist in traditional storage systems. When organizations no longer hold their data

locally, they must depend on external service providers to ensure its safety and availability. Research in this area points out that although cloud systems improve scalability and efficiency, they also create concerns related to trust, control, and long-term reliability [2].

These concerns are not just theoretical. There have been several instances where users experienced unexpected data loss in cloud-based services, highlighting the risk of depending entirely on third-party storage [3]. In other cases, service providers have shut down their platforms, leaving users without access to their stored data, which raises serious questions about data persistence and ownership [4].Even large and well-established cloud platforms are not free from failures. Storage service outages have temporarily disrupted access to user data, showing that availability cannot always be guaranteed [5]. Similarly, application-level disruptions in cloud environments have affected the continuity of services, especially for systems that depend on real-time data access [6].Apart from availability issues, security breaches have become a major concern. Unauthorized access to sensitive data, including financial and personal information, has demonstrated that cloud environments can be targeted by attackers if proper safeguards are not in place [7]. These incidents highlight the importance of not only storing data securely but also continuously verifying its integrity.To address this need, mechanisms such as Provable Data Possession (PDP) were developed. This approach allows users to check whether their data remains intact on the cloud without downloading the entire dataset, making the verification process efficient and practical [8]. Building on this idea, researchers introduced methods that support auditing while keeping the actual data hidden from the auditor, thereby improving privacy during the verification process [9]. Developments made it possible to perform auditing through a third-party entity while still allowing updates to the stored data. This means that data can be modified, inserted, or deleted without affecting the ability to verify its correctness, which is important for dynamic applications [10].Even with these advancements, maintaining privacy during auditing—especially for sensitive domains like healthcare remains a challenging task. This work focuses on addressing that gap by proposing a secure and scalable auditing approach that protects both data content and user identity while ensuring efficient verification in cloud environments.

II LITERATURE SURVEY

Cloud computing has been widely studied as a transformative technology that enables flexible and scalable data storage solutions. The foundational definition provided by Mell and Grance explains cloud computing as a model that offers on-demand access to shared computing resources, forming the basis for modern cloud systems [1]. Building on this, Armbrust et al. presented a detailed analysis of cloud computing, highlighting both its advantages, such as cost efficiency and scalability, and its challenges, including data security, reliability, and trust management [2].

Practical incidents have played a key role in shaping research directions in this field. Reports of large-scale data loss in cloud-based email services revealed the potential risks of relying entirely on cloud providers for data storage [3]. Similarly, the sudden shutdown of online storage services demonstrated that users could lose access to their data without prior notice, emphasizing the need for reliable backup and verification mechanisms [4].

Further evidence of cloud vulnerabilities came from major service outages, where storage systems became temporarily unavailable, affecting millions of users and applications [5]. Application-level disruptions in cloud platforms also showed that service reliability is a critical concern, particularly for systems that depend on continuous availability [6]. These incidents collectively

underline the importance of designing secure and dependable cloud storage systems.

Security breaches have further intensified concerns regarding cloud adoption. Incidents involving unauthorized access to sensitive financial data highlighted the weaknesses in traditional security mechanisms and the need for stronger data protection strategies [7]. These challenges have driven researchers to explore cryptographic approaches for ensuring data integrity and confidentiality in cloud environments.

One of the most significant contributions in this area is the introduction of Provable Data Possession (PDP) by Ateniese et al., which allows users to verify the integrity of outsourced data without retrieving it completely. This approach reduces communication overhead while maintaining assurance about data correctness [8]. PDP has become a foundational concept for many subsequent cloud auditing schemes.

Building upon PDP, Shah et al. proposed privacy-preserving auditing mechanisms that enable verification of data integrity without exposing the actual data content to the auditor. This work addressed one of the key limitations of earlier approaches by incorporating privacy into the auditing process [9].

Further advancements were made by Wang et al., who introduced a publicly verifiable auditing scheme that supports dynamic data operations. Their approach allows data to be updated, inserted, or deleted while still maintaining the ability to verify its integrity through a Third-Party Auditor (TPA) [10]. This development significantly improved the practicality of cloud auditing systems in real-world applications.

III RELATED WORK

The idea of storing data in the cloud has evolved steadily over the years, moving from simple storage services to complex platforms that support large-scale applications. Early work in this area mainly discussed how cloud systems reduce the burden of maintaining physical infrastructure and make data access more flexible. However, as organizations began shifting sensitive information to the cloud, it became clear that convenience alone was not enough. Questions about trust, ownership, and long-term reliability started to gain attention, especially when users realized they no longer had direct control over their data.

As cloud adoption increased, a number of real incidents exposed weaknesses in these systems. There have been situations where users suddenly lost access to their data due to service shutdowns or unexpected failures. In other cases, temporary outages disrupted access to important information, affecting both individuals and businesses. These experiences made it evident that storing data remotely requires more than just availability—it demands mechanisms to ensure that data remains safe, intact, and accessible at all times.

Security concerns further complicated the problem. With sensitive data such as personal records and financial details being stored online, the risk of unauthorized access became a serious issue. Traditional security methods were not always sufficient to handle the unique challenges of cloud environments. This led researchers to explore new ways of protecting data, focusing not only on preventing unauthorized access but also on verifying that the stored data has not been altered or corrupted.

To deal with integrity verification, several techniques were introduced that allow users to check their data without downloading it completely. These methods made the process more efficient and practical, especially when dealing with large datasets. Over time, improvements were made to ensure that the verification process itself

does not expose sensitive information. This gave rise to privacy-aware auditing approaches, where data can be verified without revealing its actual content.

Another important development was the introduction of external auditing, where a separate entity performs verification on behalf of the data owner. This reduces the workload on users and makes the system more scalable. At the same time, support for dynamic data operations—such as updating or deleting records—made these solutions more adaptable to real-world scenarios. Even with these advancements, certain limitations still remain, particularly in balancing privacy, efficiency, and simplicity.

IV PROBLEM STATEMENT

As cloud platforms become the default choice for storing medical records, a difficult trade-off begins to appear. On one hand, healthcare systems benefit from easy access, scalability, and reduced infrastructure costs. On the other hand, once patient data is moved to remote servers, the data owner no longer has direct visibility or control over how that data is stored, maintained, or protected. This creates uncertainty about whether the information remains accurate, complete, and untouched over time. A major concern lies in verifying the integrity of stored data. Healthcare data cannot afford silent corruption or partial loss, yet downloading entire datasets for verification is not practical due to size and cost. While auditing mechanisms exist to solve this problem, many of them introduce a new issue—privacy leakage. During verification, sensitive details or even the identity of the data owner may become visible to external auditors, which is unacceptable in a medical context where confidentiality is critical. Another complication arises from the complexity of existing security frameworks. Many solutions depend on traditional certificate-based systems that require continuous management of keys and certificates. This not only increases administrative overhead but also makes the system harder to scale as the number of users grows. In real-world healthcare environments, where systems must handle large volumes of data and multiple users simultaneously, such complexity becomes a serious limitation. Modern cloud systems must support frequent updates. Medical records are not static—they are continuously modified, extended, or corrected. Any practical solution must allow these dynamic operations without weakening the ability to verify data integrity. However, maintaining efficiency while supporting updates and large-scale auditing remains a challenging task

V PROPOSED SYSTEM

The system proposed in this work is designed with a clear goal in mind: to make cloud storage safe for medical data without making the process complicated or resource-heavy. Instead of relying on traditional approaches that either expose sensitive information or demand complex management, this model focuses on keeping things secure, private, and efficient at the same time.

The process begins at the data owner's side, where the medical file is prepared before it is sent to the cloud. Rather than uploading the file directly, the sensitive portions of the data are carefully hidden using a transformation technique. This ensures that personal details are not visible in their original form. At the same time, a set of verification elements is created alongside the data. These elements act like checkpoints that can later confirm whether the data has been changed or remains intact.

Once the data is prepared, it passes through a controlled processing stage that further refines the transformed content. This stage ensures that all hidden information follows a consistent structure and that the verification elements remain valid even after processing. The result is a sanitized version of the original file that can safely be stored in the cloud without revealing critical details. Even

if someone gains access to the stored file, they will not be able to interpret the sensitive parts.

To make sure the stored data remains trustworthy, the system includes an auditing mechanism. Instead of requiring the data owner to perform checks repeatedly, an external auditor is allowed to verify the data. What makes this approach different is that the auditor does not see the actual data during verification. The process relies only on the prepared verification elements, which means the privacy of both the data and the user is maintained throughout.

The system also considers practical usage scenarios. Medical data is not static, so the design supports updates such as adding new records, modifying existing ones, or removing outdated information. These operations can be performed without disturbing the auditing process. In addition, the system can handle multiple verification requests together, which helps maintain performance even when the number of users grows.

The proposed system avoids the usual trade-offs between security and usability. It protects sensitive information, allows reliable verification, and keeps the overall process simple enough to be applied in real-world healthcare environments.

VI METHODOLOGY

The approach followed in this work is built around a simple idea: data should remain protected and verifiable from the moment it leaves the owner's system until it is accessed again. To achieve this, the method does not treat security, privacy, and auditing as separate steps. Instead, all of them are woven into a single flow so that each stage naturally supports the next.

The process begins with an initial setup phase where the basic framework of the system is established. A trusted component is responsible for preparing the environment and issuing secure credentials to users based on their identity. This avoids the need for managing multiple certificates and keeps the overall system easier to handle, especially when new users are added.

Once the setup is complete, the data owner prepares the medical file before sending it to the cloud. Rather than uploading the file as it is, the sensitive parts are carefully hidden. This step ensures that private information is not exposed, even if the stored file is accessed without authorization. Along with this, a set of supporting values is created, which later helps in checking whether the data has been changed in any way. These values are tied to the content of the file, so even small modifications can be detected.

After this preparation, the data goes through an intermediate stage where it is adjusted into a consistent format. This step makes sure that the protected data and its verification elements remain aligned with each other. The output of this stage is a clean and secure version of the file that can be stored safely in the cloud. At this point, the cloud server only holds a protected form of the data, which reduces the risk of misuse.

When verification is needed, the system allows an external entity to perform the check. Instead of accessing the actual data, the auditor interacts with the cloud through a challenge–response process. The cloud generates a proof using the stored file and its associated values, and the auditor verifies this proof to confirm that the data is still intact. Since the original content is never revealed during this step, privacy is preserved throughout the process.

The method also considers practical usage where data is not static. Medical records may need to be updated, corrected, or expanded over time. The system supports these changes without breaking the verification process. In addition, it can handle multiple verification requests together, which helps maintain performance even as the system grows.

VII IMPLEMENTATION

The system is implemented in a way that keeps the design practical and easy to manage, while still meeting the security requirements of medical data storage. Instead of building everything as a single block, the application is divided into smaller modules, each handling a specific responsibility. This makes the system easier to understand, maintain, and extend when needed.

The user interface is developed as a web-based platform so that different users can access it through a browser. Separate interfaces are provided for data owners, users, auditors, and the administrative component. Each user logs in with their credentials, and their actions within the system are restricted based on their role. This ensures that only authorized individuals can perform sensitive operations such as uploading files or requesting access.

When a data owner uploads a file, the system does not store it immediately. The file is first processed to protect any sensitive information it contains. This step modifies the data in such a way that confidential details are hidden, while still keeping the file usable. At the same time, the system generates a set of verification values linked to the content of the file. These values act as a reference that will later help in checking whether the data has been changed.

After this preparation, the file is passed through a controlled processing stage that ensures consistency in how the data is stored. This stage also takes care of maintaining the connection between the processed file and its verification values. Once this step is complete, the file is uploaded to the cloud storage. The cloud server only holds the protected version of the file, which reduces the chances of sensitive information being exposed.

Access to stored data is handled carefully. When a user wants to retrieve a file, they must first request permission. The system verifies the request and, if approved, allows the user to download the file. Even then, access is controlled so that only the intended information can be viewed. This prevents unauthorized users from misusing the data.

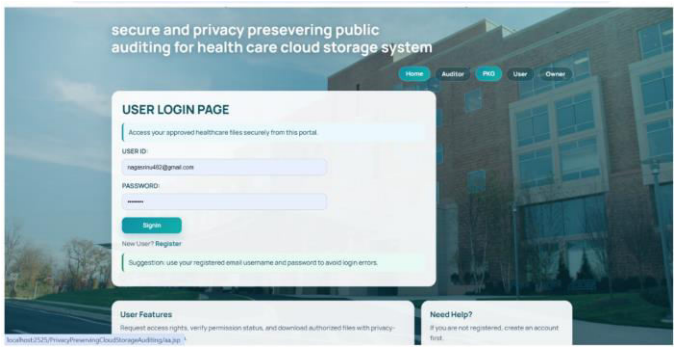
The auditing part of the system is implemented as an independent function. Instead of requiring the data owner to manually check files, an external auditor can verify the integrity of the stored data. The auditor sends a request to the cloud, and the cloud responds with a proof generated from the stored file and its verification values. The auditor checks this proof to confirm that the data is still intact. Importantly, this process does not reveal the actual content of the file, which helps maintain privacy.

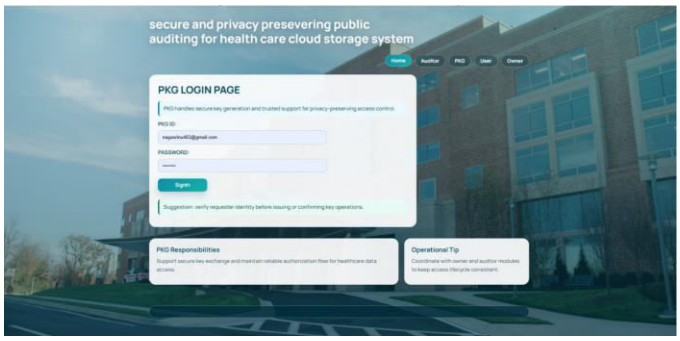
The system also supports regular updates to the data. If a file needs to be modified, added, or removed, the system updates the related verification values accordingly. This ensures that future audits remain accurate even after changes are made. Additionally, multiple auditing requests can be processed together, which improves performance when the system is handling many users at the same time.

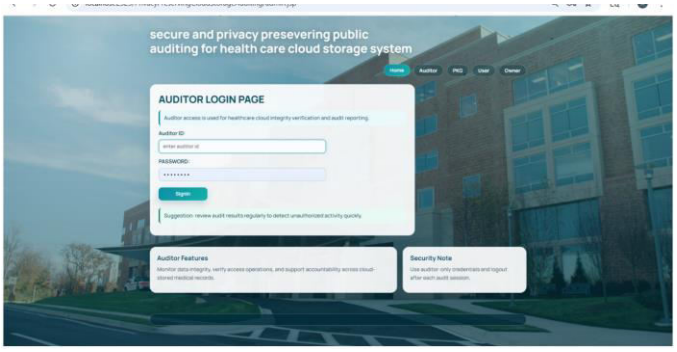
VIII RESULTS AND ANALYSIS

The system was tested with different data sizes and usage conditions to understand how it behaves in a realistic environment. The focus was not only on speed but also on how efficiently it maintains data integrity without exposing sensitive information. The results show that the approach performs consistently well, even as the size of the data grows.

One of the key observations is related to auditing time. Since the system does not rely on downloading the entire file for verification, the time required remains relatively low. As the file size increases, the auditing time also increases, but the growth is gradual rather than drastic. This makes the system suitable for handling large medical datasets without causing noticeable delays.







Another important aspect is the comparison between individual auditing and batch processing. When multiple verification requests are handled separately, the total time increases quickly. However, when the same requests are processed together, the system avoids repeated computations and reduces overall effort. This clearly shows that batch auditing improves efficiency, especially in environments where many users are active at the same time.

Communication overhead is also kept under control. Instead of transferring complete files during verification, the system exchanges only small proof values. This significantly reduces network usage and makes the system more practical in situations where bandwidth is limited. It also speeds up the interaction between the auditor and the cloud server.

The accuracy of detecting data changes is another strong point. Even small modifications in the stored data are identified with a high level of confidence. As the amount of altered data increases, the probability of detection becomes even stronger, ensuring that the system can reliably identify any unauthorized changes.

Table 1: Auditing Time for Different File Sizes

File Size (MB)	Auditing Time (ms)
10	115
50	170
100	240
200	330
500	500

Table 2: Performance of Single vs Batch Auditing

Number of Requests	Single Auditing (ms)	Batch Auditing (ms)
5	580	500
10	1150	940
20	2300	1850

Number of Requests	Single Auditing (ms)	Batch Auditing (ms)
50	5800	4300

Table 3: Communication Overhead

Operation	Data Transfer (KB)
File Upload	1024
Audit Proof	4
Verification	2

Table 4: Detection Capability

Data Altered (%)	Detection Rate (%)
1%	94%
5%	97%
10%	99%
20%	100%

The results indicate that the system achieves a balanced performance. It manages to keep the auditing process fast, reduces communication cost, and maintains strong accuracy in detecting data changes. At the same time, it preserves privacy, which is essential for handling sensitive medical information. These characteristics make it suitable for real-world cloud-based healthcare systems.

IX CONCLUSION

The work presented here takes a practical look at a problem many systems quietly struggle with—how to trust cloud storage with information that should never be exposed. In the case of medical data, even a small leak or unnoticed change can have serious consequences. That is why the focus has been on building a system that does more than just store data; it actively protects and verifies it without making the process complicated for users.What makes this approach effective is the way it handles sensitive information from the very beginning. Instead of treating privacy as an afterthought, the system ensures that confidential details are concealed before the data leaves the owner's side. This means that even if the cloud environment is not fully trusted, the most critical information remains protected. At the same time, the ability to check whether the data has been altered is preserved through a lightweight verification process that does not reveal the actual content.The system also shows that strong security does not have to come at the cost of performance. By avoiding unnecessary data transfers and supporting grouped verification requests, it manages to keep operations efficient even when the workload increases. The ability to handle updates without breaking the verification process further adds to its practicality, especially in environments like healthcare where data changes frequently.Rather than introducing complex structures, the design keeps things manageable by reducing dependence on heavy key management systems. This makes it easier to deploy and maintain, which is often a challenge in real-world applications. The result is a solution that quietly balances protection, efficiency, and usability without leaning too heavily in any one direction.

REFERENCES

[1] C. Sivapragash, S. R. Thilaga, and S. S. Kumar, "Advanced cloud computing in smart power grid," in Proc. IET Chennai 3rd Int. Sustain. Energy Intell. Syst., 2014, pp. 356–361.

[2] C. Sivapragash, S. Padmanaban, H. Eklas, J. B. Holmnielsen, and R. Hemalatha, "Location-based optimized service selection for data management with

cloud computing in smart grids," Energies, vol. 12, no. 23, 2019, Art. no. 4517.

[3] S. Kumar and C. Sivapragash, "Time orient traffic estimation approach to improve performance of smart grids," J. Comput. Theor. Nanosci., vol. 13, no. 8, pp. 5037–5045, 2016.

[4] D. G. Chandra and R. S. Bhadoria, "Role of G-cloud in citizen centric governance," in Proc. IEEE Int. Conf. Parallel Distrib. Grid Comput., 2012, pp. 44–48.

[5] K. Ren, C. Wang, and Q. Wang, "Security challenges for the public cloud," IEEE Internet Comput., vol. 16, no. 1, pp. 69–73, Jan./Feb. 2012.

[6] K. S. Jadon, R. S. Bhadoria, and G. S. Tomar, "A review on costing issues in big data analytics," in Proc. Int. Conf. Comput. Intell. Commun. Netw., 2015, pp. 727–730.

[7] R. S. Bhadoria, "Security architecture for cloud computing," Handbook of Research on Securing Cloud-Based Databases With Biometric Applications. Hershey, PA, USA: IGI Global, 2015.

8] G. Ateniese et al., "Provable data possession at untrusted stores," in Proc. 14th ACM Conf. Comput. Commun. Secur., 2007, pp. 598–609.

[9] A. Juels and B. S. Kaliski Jr, "Pors: Proofs of retrievability for large files," in Proc. 14th ACM Conf. Comput. Commun. Secur., 2007, pp. 584–597.

[10] K. Liang et al., "A DFA-based functional proxy re-encryption scheme for secure public cloud data sharing," IEEE Trans. Inf. Forensics Secur., vol. 9, no. 10, pp. 1667–1680, Oct. 2014.

[11] W. Shen, J. Qin, J. Yu, R. Hao, and J. Hu, "Enabling identity-based integrity auditing and data sharing with sensitive information hiding for secure cloud storage," IEEE Trans. Inf. Forensics Secur., vol. 14, no. 2, pp. 331–346, Feb. 2018.

[12] G. Ateniese, D. H. Chou, B. De Medeiros, and G. Tsudik, "Sanitizable signatures," in Proc. Eur. Symp. Res. Comput. Secur, 2005, pp. 159–177.

[13] H. Shacham and B. Waters, "Compact proofs of retrievability," J. Cryptol., vol. 26, no. 3, pp. 442–483, 2013.

[14] D. Boneh, B. Lynn, and H. Shacham, "Short signatures from the weil pairing," J. Cryptol., vol. 17, no. 4, pp. 297–319, 2004.

[15] J. Yu, K. Ren, and C. Wang, "Enabling cloud storage auditing with verifiable outsourcing of key updates," IEEE Trans. Inf. Forensics Secur., vol. 11, no. 6, pp. 1362–1375, Jun. 2016.

[16] J. Yu and H.Wang, "Strong key-exposure resilient auditing for secure cloud storage," IEEE Trans. Inf. Forensics Secur., vol. 12, no. 8, pp. 1931–1940, Aug. 2017.

[17] Y. Xu, S. Sun, J. Cui, and H. Zhong, "Intrusion-resilient public cloud auditing scheme with authenticator update," Inf. Sci., vol. 512, pp. 616–628, 2020.

[18] R. Ding, Y. Xu, J. Cui, and H. Zhong, "A public auditing protocol for cloud storage system with intrusion-resilience," IEEE Syst. J., vol. 14, no. 1, pp. 633–644, Mar. 2020.

[19] D. He, S. Zeadally, and L. Wu, "Certificateless public auditing scheme for cloud-assisted wireless body area networks," IEEE Syst. J., vol. 12, no. 1, pp. 64–73, Mar. 2018.

[20] D. He, N. Kumar, S. Zeadally, and H. Wang, "Certificateless provable data possession scheme for cloud-based smart grid data management systems," IEEE

Trans. Ind. Informat., vol. 14, no. 3, pp. 1232–1241, Mar. 2018.

[21] J. Li, H. Yan, and Y. Zhang, "Certificateless public integrity checking of group shared data on cloud storage," IEEE Trans. Services Comput., to be published, doi: 10.1109/TSC.2018.2789893.

[22] C. Wang, S. S. Chow, Q. Wang, K. Ren, and W. Lou, "Privacy-preserving public auditing for secure cloud storage," IEEE Trans. Comput., vol. 62, no. 2, pp. 362–375, Feb. 2013.

[23] W. Shen, J. Yu, H. Xia, H. Zhang, X. Lu, and R. Hao, "Light-weight and privacy-preserving secure cloud auditing scheme for group users via the third party medium," J. Netw. Comput. Appl., vol. 82, pp. 56–64, 2017.

[24] P. Zhao, J. Yu, and H. Zhang, "Secure outsourcing algorithm for signature generation in privacy-preserving public cloud storage auditing," J. Inf. Sci. Eng., vol. 35, no. 3, pp. 635–650, 2019.

[25] S. Anbuchelian, C. Sowmya, and C. Ramesh, "Efficient and secure auditing scheme for privacy preserving data storage in cloud," Cluster Comput., vol. 22, no. 4, pp. 9767–9775, 2019.

[26] J. Han, Y. Li, and W. Chen, "A lightweight and privacy-preserving public cloud auditing scheme without bilinear pairings in smart cities," Comput. Standards Interfaces, vol. 62, pp. 84–97, 2019.

[27] Y. Yu et al., "Identity-based remote data integrity checking with perfect data privacy preserving for cloud storage," IEEE Trans. Inf. Forensics Security, vol. 12, no. 4, pp. 767–778, Apr. 2017.

[28] B. Wang, B. Li, and H. Li, "Oruta: Privacy-preserving public auditing for shared data in the cloud," IEEE Trans. Cloud Comput., vol. 2, no. 1, pp. 43–56, Jan./Mar. 2014.

[29] G. Yang, J. Yu, W. Shen, Q. Su, Z. Fu, and R. Hao, "Enabling public auditing for shared data in cloud storage supporting identity privacy and traceability," J. Syst. Softw., vol. 113, pp. 130–139, 2016.

[30] A. Fu, S. Yu, Y. Zhang, H. Wang, and C. Huang, "NPP: A new privacy-aware public auditing scheme for cloud data sharing with group users," IEEE Trans. Big Data, to be published, doi: 10.1109/TBDATA.2017.2701347