The Indonesian Journal of Computer Science
www.ijcs.net
Volume 14, Issue 3, June 2025
https://doi.org/10.33022/ijcs.v14i3.4892

Implementing Privacy-Preserving Record Linkage to Enhance Customer Relationship Management: A Case Study at an Indonesian Manufacturing Company

Muhamad Ikbal[1], Achmad Nizar Hidayanto[2], Ni Wayan Trisnawaty[3], R. Yugo Kartono Isal [4]
Muhamad.ikbal12@ui.ac.id[1], nizar@cs.ui.ac.id[2], ni.wayan05@ui.ac.id[3], yugo@cs.ui.ac.id [4]
[1,2,3,4] Faculty of Computer Science, Universitas Indonesia, Jakarta, Indonesia

Article Information

Received : 28 May 2025
Revised : 10 Jun 2025
Accepted : 16 Jun 2025

Keywords

Privacy-preserving record linkage, CRM, hashing, data privacy, Indonesian manufacturing

Abstract

As cross-organizational collaboration increases, balancing data utility with privacy protection becomes essential. This study addresses the challenge by implementing Privacy-Preserving Record Linkage (PPRL) using a deterministic hashing approach at an Indonesian manufacturing firm and a leasing company. The system employs a dual-layer hashing technique (MD5 followed by salted SHA-256) to securely link standardized identifiers without revealing raw personal data. The objective was to enhance Customer Relationship Management (CRM) by identifying shared customers for targeted outreach. The approach yielded 2.6 million matched records out of over 36 million, enabling the leasing firm to achieve a 2-4% conversion rate through personalized campaigns. Results demonstrate high efficiency, scalability, and compliance with Indonesia's data protection law, offering a replicable framework for privacy-conscious data integration in regulated environments.

A. Introduction

Linking records across different datasets is crucial for analytics and insights in many domains (e.g., health, marketing, finance) [1], [2]. When sharing data between parties (here, two companies) involves sensitive personal information, privacy-preserving record linkage (PPRL) methods are needed so that linkage can occur without revealing raw personally identifiable information (PII). Traditional probabilistic linkage (e.g., Fellegi–Sunter models) relies on clear-text identifiers and is often not permissible under privacy laws (General Data Protection Regulation (GDPR), Health Insurance Portability and Accountability Act (HIPAA)).

This study has developed PPRL techniques to address this challenge. For example, Bloom filters enable fuzzy matching on encrypted q-gram encodings [1], [2], while tokenization or cryptographic hashing links records through exact matches on hashed identifiers. In health informatics, systems like Health Verity and the National Institutes of Health Global Unique Identifier (NIH GUID) approach use cryptographic tokens (salted hashes) to match patient records without sharing names or dates of birth.

B. Related Work

PPRL has been studied and implemented across various fields using different methods, as listed in Table 1. Schnell et al. introduced a probabilistic PPRL method utilizing Bloom filters to encode identifiers, allowing for approximate matching while preserving privacy. Their approach demonstrated linkage results comparable to non-encrypted identifiers and superior to phonetic encodings, with a low computational burden, making it suitable for various applications requiring privacy-preserving record linkage [1].

This study evaluated Generic Health Network Information Technology for Evidence (GRHANITE) software in [3] as part of the The Australian Collaboration for Coordinated Enhanced Sentinel Surveillance (ACCESS) project, where it extracts and links de-identified data from participating clinics and laboratories. The GRHANITE tool generates irreversible hashed linkage keys based on patient-identifying data captured in electronic medical records, employing probabilistic linkage principles to account for the variability and completeness of the underlying patient identifiers [3]. Advanced Bloom filter strategies, such as dynamic multi-key linkage, employ multiple hash combinations per record to simulate probabilistic flexibility without sacrificing security [1], [2]. This approach demonstrated superior performance to single-hash deterministic and Bloom filter methods in large-scale administrative datasets [4], [5].

Another method, like deterministic hashing, relies on the exact equality of transformed PII, simplifying implementation but assuming data is consistently formatted. Previous research shows that this method can be highly effective. Bian et al. created the “OneFL deduper” for a Florida health network, where each site hashed patient identifiers using salted SHA-256. A central system then linked records by matching these hashes [6]. Their tool achieved 97–99.7% precision and about 75.5% recall, highlighting that deterministic hashing offers excellent precision (few false matches), though recall may drop when data is incomplete or inconsistent.

Table 1. Studies Utilizing Privacy-Preserving Linkage Methods for Real-World Projects

Paper	Year	Country	Methodology	Application Domain
[1]	2009	Germany	Bloom filters on q-grams of PII	Population surveys, social research
[2], [5]	2020	Australia	Bloom filters	Hospital and clinical data
[3]	2020	Australia	Multiple hashed PII keys (GRHANITE)	Public health surveillance
[6]	2019	US	Salted SHA-256 on composite PII keys	Clinical research networks
[7]	2023	US	Multiple hashes of PII components	Electronic Health Records (EHRs)
[8]	2024	US	Multiple hashes of PII components	Multisite medical research

Deterministic matching is a foundational technique in PPRL that relies on the exact equivalence of selected attributes between records from different data sources. This method is particularly effective when applied to high-quality, consistently formatted PII in a privacy-preserving context. Unlike probabilistic methods that tolerate minor discrepancies, deterministic matching ensures high precision by only linking records when designated fields (e.g., name, date of birth, or identifier) match exactly. When implemented in PPRL, deterministic matching allows organizations to conduct secure record linkage without revealing raw identifiers [9].

To ensure privacy, this study performs deterministic matching on hashed data representations. A cryptographic hash function is applied to each selected identifier, often incorporating a secret encryption key to prevent reverse engineering. This one-way function guarantees that the same input will always produce the same output, and the result cannot be inverted to expose the original PII. For example, hashing a record's first name, last name, and date of birth with a secret key ensures that only entities sharing the same input values and encryption configuration can produce the same hash, allowing them to identify matches while preserving privacy [9].

The process may involve generating multiple hashed values per record using different combinations of attributes. This multi-hash approach enhances match reliability by increasing the likelihood of identifying valid matches across various linkage configurations. The PPRL strategy report shows that typical implementations of hash combinations such as FirstName+State+InsuranceNumber or LastName+DOB+State, each concatenated with an encryption key, produce unique and secure match keys [9].

To enhance robustness, some deterministic matching methods incorporate multiple hashing strategies or combine hash types, such as SHA-256 and HMAC (Hash-based Message Authentication Code). This layering protects common attack vectors like frequency and dictionary attacks, which attempt to infer original values from hashed outputs. Salted or keyed hashes make the output unique to the data partner's environment and resist correlation, even when the input data is identical [9].

Deterministic matching has shown exceptional effectiveness in practical applications. Bian et al. introduced OneFL Deduper, an open-source tool that uses exact hash-matching on multiple PII combinations for privacy-preserving linkage of millions of healthcare records, achieving precision above 97% across six providers [6].

The deterministic method, however, requires stringent consistency across all participating datasets. Minor differences such as formatting discrepancies, additional white space, or abbreviations can prevent an exact match. This condition necessitates rigorous data preprocessing, including cleaning, standardization, and harmonization of formats. According to the MITRE Corporation report, the quality and coordination of these steps across data partners significantly influence the final linkage quality and privacy assurance [9].

Despite these limitations, deterministic matching using hashed exact-match keys remains a preferred approach in environments where data fidelity is high, and the operational need for simple, auditable matching logic outweighs tolerance for data variability. Studies across healthcare, government, and administrative data settings consistently demonstrate that deterministic PPRL methods provide substantial linkage precision, ease of implementation, and compliance with privacy mandates [4].

This work focuses on a deterministic, exact-match approach using salted hashing. Two companies (manufacturing and leasing) agree to share only *product-level* data for everyday items and link customer records behind the scenes. To protect privacy, each party hashes all PII fields before any exchange. This study deploys a linkage architecture where each company preprocesses and hashes its PII locally and then sends the hashes to a secure matching server. The matching agent performs exact comparisons of hashes (composite name+DOB keys) to identify familiar individuals. This design follows the separation principle used in data linkage units: PII is transformed into a "linkage key" before any matching [3], [4].

The contributions of this study are: (1) the design of the PPRL system architecture for inter-company linking; (2) the data cleansing and dual-hash methodology (UPPER(MD5) + salt→SHA256) with detailed pseudocode; (3) an evaluation of performance metrics (match rate, false positives/negatives, processing time) in a real deployment; (4) discussion of scalability and future work.

C. Research Method

This study implements a PPRL method using deterministic hashing to match customer data across organizational boundaries securely. This study chose deterministic hashing with SHA-256 and MD5 + Salt, as this approach provides high performance, strong one-way encryption, and minimal risk of false positives when data is pre-standardized. This method is particularly suitable for environments where exact matches on PII, such as name and date of birth, are feasible after cleansing.

The system will involve two data custodians (Company A and Company B) and a secure matching service (which could be a trusted third party or one of the companies under an agreement). Figure 1 illustrates the architecture.

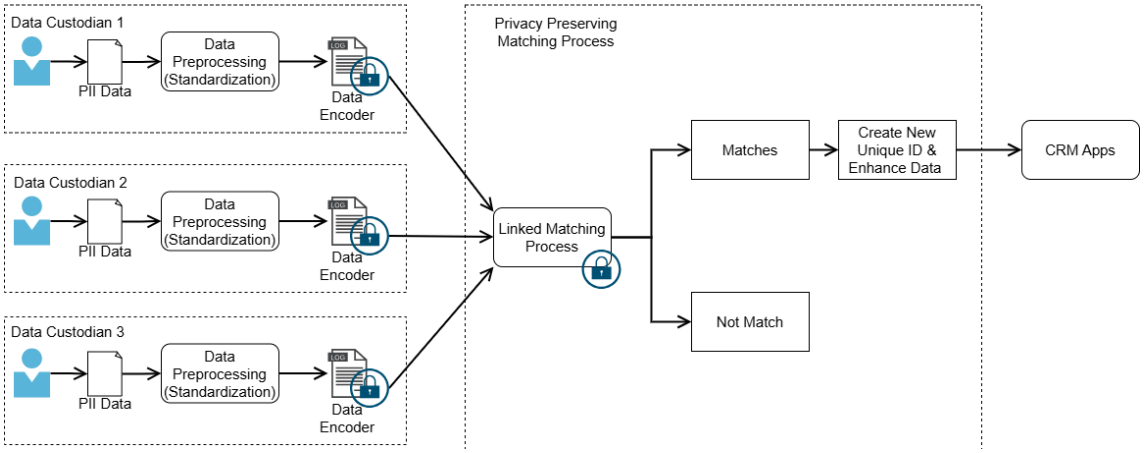

Figure1. Privacy-Preserving Data Linkage Flow

This architecture mirrors previous hash-based PPRL systems: for example, the OneFL Deduper setup where sites A and Bun a *Hasher* produce salted SHA-256 hashes and send them to a *Linker* at the data coordinating center [6]. In our case, the matching server plays the role of the linker. The separation of roles and the use of salt agree with best practices: by design, no single party can reverse the hashes without the salt, and any linkage (matching hash) can only be interpreted as a private pairing of de-identified records[3].

Using this architecture, the PPRL process follows six distinct implementation stages described below. Figure 2 illustrates the general framework of the proposed workflow.

1. Establishing Data Transfer and Data Use Agreements (DSA and DUA) between data custodians to define governance, data fields, and permitted usage.
2. Extracting and preprocessing data to standardize formats and remove inconsistencies.
3. Generating secure hash keys from composite identifiers
4. Exchanging de-identified, hashed data between parties via a neutral third party.
5. Performing hash-based matching to identify linked records.
6. Enhance data based on mutual agreement from the data custodian.

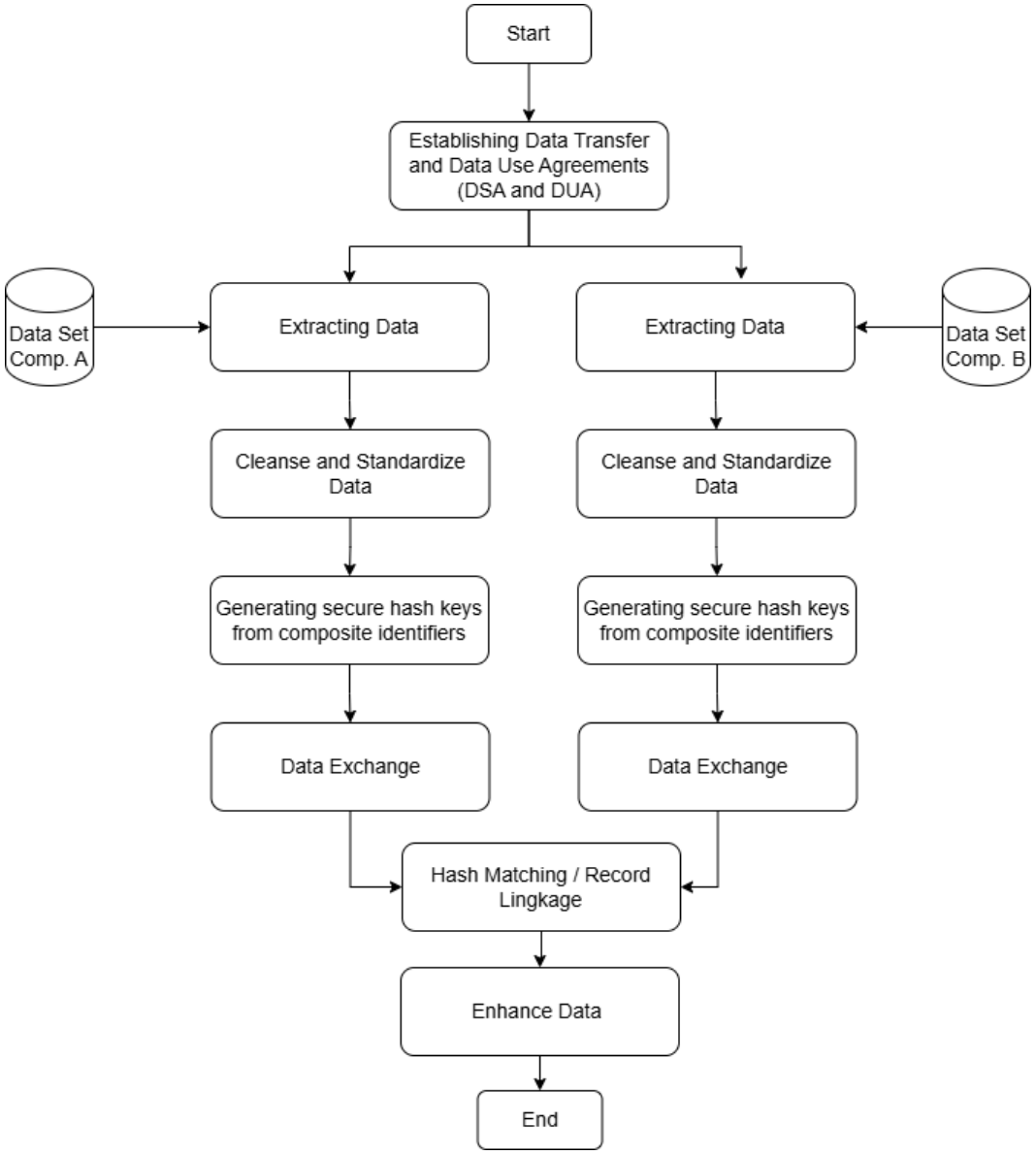

Figure2. General Framework

1. Establishing Data Transfer and Data User Agreements

Each organization's C-level executives (e.g., Chief Data Officers) sign a formal Data Sharing Agreement (DSA) to govern the data collaboration between the manufacturing and leasing companies. The DSA specifies roles, responsibilities, permitted data elements, and cryptographic standards and is aligned with best practices in privacy-preserving data integration, as seen in other real-world PPRL projects [3], [6].

A neutral third party acts as a trusted linkage agent, coordinating data preparation, receiving hashed datasets, and executing the record linkage process. All parties agree to apply a standardized hashing protocol MD5 followed by SHA-256 with a shared salt before data transfer. This approach reflects secure one-way hashing schemes adopted in clinical research networks and privacy frameworks [1], [2].

The DSA includes:

a. Organizations may exchange only product-level metadata and hashed PII (name, date of birth, address, phone, email).
b. Predefined schema: Metadata includes field names, categories, formats, record volume, and update frequency.
c. Transfer protocol: Data is transmitted over secure encrypted channels (e.g., Secure File Transfer Protocol (SFTP)/Hypertext Transfer Protocol Secure (HTTPS)) and reviewed for schema compliance before linkage.

A separate DUA governs access to the linked data. This structure mirrors models such as the NIH National COVID Cohort Collaborative (N3C) [10] that even non-contributing organizations may access insights if they meet governance and ethical standards.

Organizations must tie each request for data access to an approved business use case (e.g., propensity modeling or scoring) and subject it to internal review and audit controls. The third-party maintains only the minimum data necessary for linkage and discards hash post-processing, maintaining zero retention of sensitive data. This dual-agreement model (DSA + DUA) has proven effective in balancing privacy with cross-organizational collaboration [6], [10].

2. Data Extraction and Preprocessing

Each Company extracts customer records containing PII and product data. Before linking the records, they apply a standardized preprocessing pipeline to ensure consistency across datasets, which is critical due to the sensitivity of cryptographic hash functions to minor input differences [6].

Normalization includes converting fields to uppercase, removing punctuation, diacritics, whitespace, and common prefixes/suffixes (e.g., "Mr.," "Inc."). For example, "Acme Inc." becomes "ACME," and "O'Neil" is standardized as "ONEIL." The system reformats dates to YYYYMMDD and strips non-digit characters from phone numbers before standardizing them.

The system normalizes the full name and date of birth, constructs a composite key, and then passes it through a hash function. This key design balances accuracy and privacy, especially when ID numbers are incomplete or unreliable. This process aligns with standard PPRL practices. While tools like GRHANITE also apply phonetic encoding [3], this implementation relies on strict normalization and exact matching, as illustrated in Tables 2 and 3.

Table 2. Sample Data Input

Field	Value
Full Name	Ir. Budi Santoso, PhD
Date of Birth	Wednesday, 7 February 1960

Table 2 presents the initial raw data collected for the record linkage process. The information includes the full name "Ir. Budi Santoso, PhD" and the date of birth "Wednesday, 7 February 1960." At this stage, the data may contain inconsistencies such as honorifics and formatting variations in dates and capitalization, which could hinder effective matching across disparate data sources.

Table 3. Standardization Process for Composite Key and Expected Result

Standardization Step	Value
Remove Honorifics & Titles	Budi Santoso
Convert to Uppercase	BUDISANTOSO
Format Date of Birth (YYYYMMDD)	19600207
Concatenated Name + DOB	BUDISANTOSO19600207

Table 3 illustrates the step-by-step standardization process of the raw data to create a clean and consistent composite key. Initially, honorifics and academic titles are removed from the full name, yielding "Budi Santoso." The name is then converted to uppercase, resulting in "BUDISANTOSO" to eliminate case sensitivity issues. The date of birth is reformatted into the standardized "YYYYMMDD" format, producing "19600207." The system concatenates the standardized name and date of birth to form the composite key 'BUDISANTOSO19600207,' ensuring secure record linkage across systems.

3. Hash Key Generation

After cleansing the data, the system concatenates each record's key fields (full name and date of birth) into a composite string (e.g., NAMEYYYYMMDD). This string undergoes a two-layer deterministic hashing process to ensure speed and security. First, the system hashes the composite key using MD5. Then, it appends a shared secret salt (agreed upon by both companies) to the MD5 output and hashes the result again using SHA-256. This layered approach combines the speed of MD5 with the cryptographic strength of SHA-256, while the salt prevents precomputed attacks such as rainbow tables [3].

The following pseudocode formalizes the process.

```
pgsql
function generate_hash(record):
    composite = UPPERCASE (record.name) + "|" + record.dob
    md5_val = MD5(composite) // first hash (128-bit)
    final_hash = SHA256(UPPERCASE (md5_val) + salt) // second hash
with salt (256-bit)
    return final_hash
```

Each company's Hasher module executes this function, generating (LocalID, final_hash) pairs and securely transmitting them to a neutral third party for linkage. To maximize efficiency, the system parallelizes hashing across records. It does not retain any unhashed PII or intermediate keys after processing, ensuring privacy. After the hashing process, the system purges raw PII, exchanging only the final hash values. Tables 4, 5, and 6 illustrate the step-by-step transformation of original inputs into secure hashed outputs.

Table 4. First Hashing Process Using MD5 and Expected Result

Input	Hashing Method	Output Hash
BUDISANTOSO19600207	MD5	38c43a0beec2e1b04a294dba1c410ce5

Table 4 demonstrates the initial cryptographic transformation applied to the standardized composite key "BUDISANTOSO19600207." The MD5 hashing algorithm converts the input into a fixed-length hash string. This process generates the output hash "38c43a0bece21b04a294dbac1410ce5," which anonymizes the personal data while maintaining a consistent identifier for matching purposes.

Table 5. Adding Salt to Uppercase MD5 Hashing and Expected Result

Component	Value
Salt (18-char alphanumeric)	9F6XT1K4Z8RM2C3HPQ
Upper Case Concatenated MD5 + Salt	38C43A0BEEC2E1B04A294DBA1C410CE59F6XT1K4Z8RM2C3HPQ

Table 5 shows the introduction of a cryptographic salt, "9F6XTK1K48ZRM2C3HPQ," to enhance hash uniqueness and resist dictionary attacks. The system converts the MD5 hash to uppercase, concatenates it with the salt, and ensures that the resulting hash remains unique unless the same salt applies. Salting protects against rainbow table attacks, as noted in PPRL studies [3].

Table 6. Adding Salt to Uppercase MD5 Hashing and Expected Result

Input String	Hashing Method	Final Hash Output
38C43A0BEEC2E1B04A294DBA1C410CE59F6XT1K4Z8RM2C3HPQ	SHA-256	9c6e01d63489973ad95301f15e21257d95c7a328d0978c4b865f2fa5ddbbccf0

Table 6 applies SHA-256 to the salted MD5 string, producing an irreversible hash output:
"9c6e01d63489973ad95301f15e21257d95e73a2809708c4b8652fda5ddbbecf0"
The robust algorithm, combined with MD5 and salting, prevents the final output from being traced back to original identifiers, supporting privacy-preserving record linkage.

4. Data Exchange

Each company generates a file of (localID, hash) pairs for all records (along with any non-PII product attributes to be shared). The system transmits these hash files to the linkage server over a secure channel (e.g., SFTP or API with TLS). Note that no raw PII or recognizable strings ever leave the company servers. Flowing models like OneFlorida, each partner securely transmits its salted-hash output to a central coordinator [4].

5. Hash Matching

The linkage process begins by comparing final hash values. The server performs an inner join on the final_hash field from both parties. Any match indicates a link between records from Company A and Company B. Since 256-bit hash

collisions are unlikely, the system assumes that identical hashes indicate the same individual. The server outputs a mapping of linked IDs for downstream use, ensuring no raw identifiers are shared. If a hash from one side has multiple identical entries, this suggests duplicate records or a collision, which is flagged but typically assumed to be a non-issue due to the vast hash space.

This process is efficient, with hashing at O(N) complexity, enabling hash-join operations to complete in seconds. The system uses Apache Spark for large-scale datasets to perform distributed processing across multiple nodes, ensuring scalable performance [11]. A similar setup in the OneFlorida project successfully linked 3.5 million records using salted SHA-256 hashes, demonstrating the efficiency of this method for large-scale PPRL systems [6].

Such practices, demonstrated by the OneFlorida health network, prove that enhanced data after linkage can significantly improve business outcomes while maintaining data privacy and security [6], [12].

6. Enhancing Data

After completing the linkage process, data custodians integrate linked customer records with product data and customer interactions for CRM based on mutual agreements. The enhanced dataset helps identify targeted marketing segments and personalized engagement strategies.

This process relies on shared data definitions and agreed-upon attributes, ensuring compliance with data-sharing agreements (DSA/DUA). Post-linkage, advanced analytics, and machine learning models can be applied to predict customer behavior and prioritize outreach, leading to more effective CRM campaigns and improved conversion rates [13].

D. Result and Discussion

Two entities in the automotive and financial sectors implemented PPRL. Company A, a manufacturing firm with a customer base of 31 million, and Company B, a leasing company with 5 million records. The process followed strict privacy protocols, including a DSA and a DUA, signed at the executive level (Chief Officer) to ensure data governance and legal compliance.

The linkage utilized deterministic matching based on a composite key consisting of a standardized full name and date of birth (DOB), hashed using MD5 with salt and an additional SHA256 layer. The data standardization included cleaning titles or business suffixes, uppercasing all personal data, formatting dates, phone numbers, and emails, and removing special characters and excessive spaces.

Company A processed 31 million records in approximately 2 hours, while Company B hashed 5 million in about 48 minutes. The matching process was conducted using an Apache Spark cluster with at least 4 VCore CPU, 32 GB RAM, and 3-5 executors, depending on the Spark configuration of each company. This setup allowed the matching process to be completed in about 12 minutes, demonstrating the efficiency of distributed computing for large-scale PPRL tasks [11].

The final linkage process yielded around 2.6 million matched records, reflecting successful matches between the two datasets. Invalid data prevented approximately 1% of Company B's customer records from linking due to inconsistencies in name formatting, missing DOB values, or non-overlapping

customer bases. This outcome is comparable with similar deterministic approaches reported in other studies, such as the OneFlorida network, where a similar salted SHA-256 approach achieved precision rates of 97–99.7% but lower recall due to data quality issues [6].

From the matched records, approximately 65% were identified as potential leads, representing customers who met the initial criteria for further engagement. Company B then applied additional filtering based on key indicators such as nearing loan maturity and high credit scores. This segmentation strategy allowed Company B to prioritize around 20,000 to 30,000 leads each month for targeted CRM outreach, focusing on prospects with higher conversion potential.

By personalizing outreach based on these criteria, Company B could offer tailored products, such as loan refinancing options, vehicle upgrades, or credit limit extensions, which are particularly relevant for customers nearing the maturity of their current loans. The subsequent marketing campaign achieved a 2% to 4% conversion rate, demonstrating the significant potential for upselling or cross-selling new vehicle financing products. These results highlight the tangible business value of cross-organizational PPRL when integrated with data-driven CRM strategies, enabling personalized engagement while maintaining privacy and compliance.

The deterministic PPRL strategy used here ensures compliance with Indonesia's Personal Data Protection Law (UU PDP) by eliminating the retention of plain PII, which is crucial in cross-company collaborations. This method provides a secure, privacy-preserving framework for linking data across organizations without exposing sensitive personal identifiers. Unlike probabilistic linkage methods (e.g., Bloom filters), deterministic hashing ensures that the data remains anonymized by relying solely on hashed identifiers. This approach prevents any potential re-identification, significantly reducing the risk of privacy violations. This approach supports efficient CRM processes and facilitates data-sharing partnerships in regulated industries, ensuring compliance with personal data protection rules during collaboration.

This method offers high precision, essential for downstream applications such as customer relationship management (CRM), where false positives are costly. Unclean or missing source data values caused a 1% mismatch rate, a standard limitation in multiparty linkage scenarios.

The following elements were central to the project's success:

1. A neutral third party manages the PPRL process while retaining only minimal, temporary hashed data.
2. A robust DSA and DUA framework, defined in a standardized PDF contract signed by executive-level stakeholders.
3. Agreement on hashing standards and field preprocessing across data custodians, enabling consistent linkage logic.
4. The potential for reusability of the methodology across diverse digital ecosystems.

Significant ethical risks remain despite the use of hashed data in PPRL systems. The primary concern is the potential for re-identification. Hashing techniques like

salted SHA-256 protect personal information, but advanced methods could still enable de-anonymization if auxiliary data is available [12], [13]. Individuals often struggle to fully understand how organizations share and use their data, making informed consent a major ethical challenge. This lack of transparency can lead to misuse of personal information, such as targeted marketing based on sensitive attributes like credit scores or health data [13]. Data custodians must ensure clear communication and responsible data handling practices to avoid exploitation.

Organizations must implement strong consent mechanisms to ensure individuals fully understand how their data will be used, effectively mitigating these risks. Incorporating additional privacy-preserving methods, such as differential privacy or secure multiparty computation (SMPC), can help reduce the likelihood of data misuse and ensure compliance with privacy laws [13]. Furthermore, establishing regulatory oversight and ethical frameworks will provide a foundation for responsible cross-organizational data sharing, preventing the potential misuse of data and ensuring that privacy protection rules are upheld throughout the PPRL process [11], [12].

E. Conclusion

The use of salted deterministic hashing for privacy-preserving record linkage between two distinct organizations has proven technically effective and operationally feasible. The system minimizes data, complies with national data privacy regulations, and provides tangible business value via CRM activation and cross-sell opportunities.

Limitations include:

1. Data quality dependency: Deterministic methods are sensitive to formatting inconsistencies or missing fields.
2. Scalability concerns: While effective in a two-party setup, extending the framework to multi-lateral systems will require scalable infrastructure and standardization policies.
3. Incomplete or incorrectly formatted data leads to missed matches, offsetting the high precision—as seen in the OneFlorida use case [6].

Future work may explore hybrid models combining deterministic and fuzzy matching under secure multiparty computation (SMPC) or homomorphic encryption, as demonstrated in work by Schnell et al. and Nguyen et al., to improve recall while retaining privacy [1], [3].

F. References

[1] R. Schnell, T. Bachteler, and J. Reiher, "Privacy-preserving record linkage using Bloom filters," *BMC Med Inform Decis Mak*, vol. 9, no. 1, 2009, doi: 10.1186/1472-6947-9-41.

[2] S. Randall *et al.*, "A blinded evaluation of privacy preserving record linkage with Bloom filters," *BMC Med Res Methodol*, vol. 22, no. 1, Dec. 2022, doi: 10.1186/s12874-022-01510-2.

[3] L. Nguyen *et al.*, "Privacy-preserving record linkage of de-identified records within a public health surveillance system: Evaluation study," *J Med Internet Res*, vol. 22, no. 6, Jun. 2020, doi: 10.2196/16757.

[4] S. M. Randall, A. M. Ferrante, J. H. Boyd, J. K. Bauer, and J. B. Semmens, "Privacy-preserving record linkage on large real world datasets," *J Biomed Inform*, vol. 50, pp. 205–212, 2014, doi: 10.1016/j.jbi.2013.12.003.

[5] S. Randall, A. Brown, A. Ferrante, J. Boyd, and S. Robinson, "Implementing privacy preserving record linkage: Insights from Australian use cases," *Int J Med Inform*, vol. 191, Nov. 2024, doi: 10.1016/j.ijmedinf.2024.105582.

[6] J. Bian *et al.*, "Implementing a hash-based privacy-preserving record linkage tool in the OneFlorida clinical research network," *JAMIA Open*, vol. 2, no. 4, pp. 562–569, Dec. 2019, doi: 10.1093/jamiaopen/ooz050.

[7] K. Marsolo *et al.*, "Assessing the impact of privacy-preserving record linkage on record overlap and patient demographic and clinical characteristics in PCORnet®, the National Patient-Centered Clinical Research Network," *Journal of the American Medical Informatics Association*, vol. 30, no. 3, pp. 447–455, Mar. 2023, doi: 10.1093/jamia/ocac229.

[8] U. Tachinardi *et al.*, "Privacy-preserving record linkage across disparate institutions and datasets to enable a learning health system: The national COVID cohort collaborative (N3C) experience," *Learn Health Syst*, vol. 8, no. 1, Jan. 2024, doi: 10.1002/lrh2.10404.

[9] S. J. Petersen, R. D. Lieberthal, K. J. Miller, and N. H. Vakil, "Privacy Preserving Record Linkage (PPRL) Strategy and Recommendations Sponsor: National Institute on Aging PPRL Linkage Strategies Report McLean, VA," 2023. [Online]. Available: https://www.alz.org/alzheimers-dementia/facts-figures.

[10] M. A. Haendel *et al.*, "The National COVID Cohort Collaborative (N3C): Rationale, design, infrastructure, and deployment," *Journal of the American Medical Informatics Association*, vol. 28, no. 3, pp. 427–443, Mar. 2021, doi: 10.1093/jamia/ocaa196.

[11] O. Valkering and A. Belloum, "Privacy-Preserving Record Linkage with Spark." [Online]. Available: https://aws.amazon.com/emr/

[12] V. M. Shelake and N. M. Shekokar, "SMSPPRL: A similarity matching strategy for privacy preserving record linkage," in *PDGC 2020 - 2020 6th International Conference on Parallel, Distributed and Grid Computing*, Institute of Electrical and Electronics Engineers Inc., Nov. 2020, pp. 481–485. doi: 10.1109/PDGC50313.2020.9315828.

[13] Y. Chi, J. Hong, A. Jurek, W. Liu, and D. O'Reilly, "Privacy preserving record linkage in the presence of missing values," *Inf Syst*, vol. 71, pp. 199–210, Nov. 2017, doi: 10.1016/j.is.2017.07.001.