Breaking BAD? Better Call SAUL! – Breaking and Fixing Bloom Filters with Added Diffusion

Frederik Armknecht
University of Mannheim
armknecht@uni-mannheim.de

Jochen Schäfer
University of Mannheim
jochen.schaefer@uni-mannheim.de

Abstract

Merging records from two databases by comparing quasi-identifiers such as names or addresses is a frequently occurring task in many academic and administrative domains. Privacy-Preserving Record Linkage (PPRL) techniques aim to provide a way of computing the similarities between quasi-identifiers without revealing the plaintext data. In practice, PPRL is usually performed by applying a similarity-preserving encoding to the data and comparing similarities on the encoded data only. However, recent research demonstrated that all standard PPRL encoding schemes, with the exception of Bloom filters with added diffusion (BAD), are vulnerable to Graph Matching Attacks and should no longer be considered secure.

In this paper, we identify two properties of BAD that leak information about the plaintext to an attacker. We then proceed to show that these leaks make BAD vulnerable to an adapted variant of graph matching attacks. The newly proposed *Homomorphism-based Graph Matching Attack* is capable of re-identifying up to 91.6% of BAD-encoded records, thereby breaking the only remaining secure encoding scheme.

As a remedy, we present a new Scheme for Anonymous, Utility-preserving Linkage (SAUL), which is not only robust against our new attack and all previous ones, but also outperforms the state of the art in terms of linkage quality.

Keywords

privacy-preserving record linkage, graph matching attacks, bloom filters, homomorphism

1 Introduction

Record linkage, the task of identifying records in different databases that refer to the same individual, is a common challenge in data processing. While this task is straightforward when unique per-person identifiers are available, the challenge becomes more complex in their absence. In such cases, quasi-identifiers like names, addresses, and birthdates are typically used as substitutes [3]. This practice relies on the assumption that these attributes are stable, meaning their values do not change frequently, allowing them to serve as long-term identifiers.

However, these attributes can change over time or may be misspelled in some records. To address this issue, probabilistic record linkage frameworks, such as the foundational Fellegi-Sunter model [5], were developed. In these schemes, records are linked if the probability that they refer to the same entity, often quantified by the similarity of their quasi-identifiers, exceeds a certain threshold. The reliance on plaintext data, however, becomes highly problematic in scenarios where an individual's presence in a database is itself sensitive information, such as in medical studies or prison registries.

Although organizational measures, like employing a trusted third party for linkage, offer some protection, cryptographic schemes that hide the plaintext might be preferable [3]. This challenge motivates the field of Privacy-Preserving Record Linkage (PPRL). While PPRL can be achieved through interactive protocols like Secure Multi-Party Computation, practical applications predominantly use a non-interactive setting, meaning the original data holders are not involved in the linkage process itself[1]. Instead, each data holder applies a *similarity-preserving* encoding to their data and submits the result to a linkage unit (LU). The LU then identifies matching records based on the similarity of their encodings. For this to be secure, it must be computationally infeasible to infer the plaintext from the encoded records.

PPRL is widely used in practice, particularly in jurisdictions where legislation mandates such privacy-enhancing approaches [3]. Notable examples include:

- **USA**: Kho et al. [7] proposed a distributed, HIPAA-compliant application for linking health databases. Their system linked around seven million electronic health records from six organizations using SHA-based hashes of quasi-identifiers.
- **Europe**: Encoding and matching schemes relying on Bloom filters (BFs) were developed for a national cohort study in Switzerland [15] and for linking medical records in Germany [10].
- **Australia**: The Linking Up and Mapping Of Systems (LUMOS) project used a BF-based PPRL scheme to link health records of over one million patients from more than 100 healthcare providers [4].
- **Brazil**: A large-scale application of PPRL was employed to link official records of around 114 million citizens using a BF-based encoding scheme [12].

As these examples illustrate, real-world PPRL schemes primarily rely on BFs for encoding data, as they offer an effective trade-off between efficiency and linkage quality [3]. However, BFs are vulnerable to various known attacks, including frequency attacks [3, 8, 11, 20]. In response, alternative encoding algorithms such as tabulation MinHash (TMH) [17], two-step hashing (TSH) [13], and match-key encoding [14] were developed to address these weaknesses.

[1] In the remainder of this work, when we refer to privacy-preserving record linkage (PPRL), we are considering only non-interactive schemes.

Proceedings on Privacy Enhancing Technologies 2026(2), 6–19

https://doi.org/10.56553/popets-2026-0033

However, neither of these schemes provided sufficient security: Match-key encodings were broken within a year of their publication [19] and another, more fundamental threat emerged with the graph matching attacks (GMAs). This type of attack can re-identify plaintexts from encodings in most existing schemes, including BF, TMH, and TSH [16]. As a consequence, the vast majority of non-interactive encoding schemes have been broken to the point that they are no longer considered secure. In fact, the only known scheme demonstrating some resistance against GMAs are Bloom Filters with added diffusion (BADs), introduced in [1].

In this paper, we demonstrate that BADs exhibit an approximate additive homomorphism, meaning the XOR of two encodings is a noisy approximation of the encoding of their plaintext symmetric difference. Additionally, the hamming weight of a BAD encoding is closely correlated to the length of the plaintext it represents. We argue that this constitutes a substantial security flaw, as an attacker can exploit this structural information to re-identify records in an encoded database. To substantiate this claim, we propose a novel attack called homomorphism-based graph matching attack (H-GMA), which operates under the standard GMA attacker model. Our experimental evaluation shows that this attack is capable of re-identifying up to 91.6% of records encoded with BAD. Finally, we present the Scheme for Anonymous Utility-preserving Linkage (SAUL), a new encoding scheme specifically designed to prevent this information leakage and thereby thwart the H-GMA. Extensive theoretical and empirical analysis of SAUL shows that it is on par with BAD in terms of performance and even outperforms its linkage quality, making it a secure and practical drop-in replacement.

The paper is structured as follows: Section 2 provides the preliminaries, including an adapted description of the BF and BAD schemes. Section 3 introduces a security model. In Section 4, we describe several properties of BF and BAD, and then explain and analyze a new attack that exploits these. As a consequence, we introduce and analyze a new PPRL scheme in Section 5. Section 6 concludes the paper.

2 Preliminaries

2.1 Notation

$\mathbb{F}_2$ denotes the binary field and $\mathbb{F}_2^\ell$ the vector space of dimension ℓ over $\mathbb{F}_2$ with $\mathbf{0}$ being the all-zero vector. For a vector $\boldsymbol{V} \in \mathbb{F}_2^\ell$, $\boldsymbol{V}[i]$ denotes its i^{th} component, where $1 \leq i \leq \ell$. The Hamming weight $\mathrm{HW}(\boldsymbol{V}) = \#\{i|\boldsymbol{V}[i] \neq 0\}$ of a vector is the number of its non-zero entries. For two vectors $\boldsymbol{V}, \boldsymbol{V'}$ of the same dimension, their Hamming distance is defined as $\mathrm{HD}(\boldsymbol{V}, \boldsymbol{V'}) = \mathrm{HW}(\boldsymbol{V} \oplus \boldsymbol{V'})$. For a random variable X over $\mathbb{F}_2$, we denote its probabilities of being zero or one as $p_0(X) = Pr\,[X = 0]$ and $p_1(X) = 1 - p_0(X)$, respectively. If X is clear from the context, we omit it and simply refer to the probabilities as p_0 and p_1.

For sets $S_1, \ldots, S_k$, we define their symmetric difference as

$$\Delta(S_1, \ldots, S_k) = \{x \in S_1 \cup \ldots \cup S_k || \{j | x \in S_j\}| \text{ odd}\}. \tag{1}$$

That is, $\Delta(S_1, \ldots, S_k)$ contains all elements of $S_1 \cup \ldots \cup S_k$ that are elements of an odd number of sets S_i. An important property is that for sets of binary vectors $S_1, \ldots, S_k \subset \mathbb{F}_2^\ell$, it holds that

$$\bigoplus_{j=1}^{k} \bigoplus_{\boldsymbol{V} \in S_j} \boldsymbol{V} = \bigoplus_{\boldsymbol{V} \in \Delta(S_1, \ldots, S_k)} \boldsymbol{V}. \tag{2}$$

Moreover, for a sequence of binary values $X = (x_1 \ldots, x_n)$ with $x_i \in \mathbb{F}_2$, the majority function is defined as

$$\mathrm{maj}(X) = \begin{cases} 1, & |\{i|x_i = 1\}| > |\{i|x_i = 0\}| \\ 0, & \text{else} \end{cases} \tag{3}$$

This is extended to a sequence of vectors $V_1, \ldots, V_n \in \mathbb{F}_2^\ell$ by applying it index-wise. That is $V = \mathrm{maj}((V_1, \ldots, V_n))$ with $V[i] = \mathrm{maj}((V_1[i], \ldots, V_n[i]))$ for $i = 1, \ldots, \ell$.

2.2 Privacy-Preserving Record Linkage

The goal of record linkage is to identify records R in different databases that refer to the same individual. The basis for the latter is a binary similarity function $\mathsf{Link}_{\text{plain}}$, where $\mathsf{Link}_{\text{plain}}(R, R') = 1$ if and only if the two plain records R and R' should be matched. The aim of PPRL is to realize the same functionality on protected records, i. e. without disclosing the identities of the affected individuals:

Definition 1 (PPRL schemes). *A PPRL scheme (with respect to* $\mathsf{Link}_{\text{plain}}$*) consists of three algorithms* (Setup, Enc, Link)*:*

- Setup *takes some input parameters* Π_{in} *and generates secret parameters* Π_{sec} *and public parameters* Π_{pub} *with* $\Pi_{in} \subseteq \Pi_{pub}$*. All remaining algorithms implicitly take* Π_{pub} *as input.*
- $\mathsf{Enc}(\Pi_{sec}, R)$ *takes the secret parameters* Π_{sec} *and a plain record* $R \in \mathcal{R}$ *and outputs an encoded record* $E(R)$*.*
- $\mathsf{Link}(E(R), E(R'))$ *takes two encoded records* $E(R)$ *and* $E(R')$ *and outputs a bit* $b \in \{0, 1\}$*.*

The idea is that instead of deciding whether two plain records R and R' shall be linked, this can be accomplished by Link on the respective encoded records. Of course, this only makes sense if the encoding process Enc preserves similarity, that is

$$\mathsf{Link}\,(\mathsf{Enc}(\Pi_{sec}, R), \mathsf{Enc}(\Pi_{sec}, R')) \;=\; \mathsf{Link}_{\text{plain}}(R, R') \tag{4}$$

with high probability.

In practice, $\mathsf{Link}_{\text{plain}}(R, R')$ internally computes a *similarity score* $\mathsf{sim}_{plain}(R, R') \in [0, 1]$ and outputs 1 if the similarity score exceeds a threshold τ. Analogously, Link computes a similarity score sim on encoded records and likewise outputs 1 if the score exceeds a threshold τ_{enc}.

Common choices for sim are set similarity measures such as the Jaccard or Dice coefficient calculated on their q-gram sets. These are the sets of all sequences of q successive characters generated from the strings representing the records. For example, if the record is *JonDoe* and $q = 2$, this would result in $\{Jo, on, nD, Do, oe\}$. Encoded records commonly are fixed-length bitstrings, meaning that sim could also be their Hamming distance.

2.3 Bloom filters and Bloom Filters with added diffusion

The very popular BF encoding scheme, as well as its extension BAD, both conform to Definition 1. They operate over records that are subsets of $\mathcal{R}$. That is, $\mathcal{R}$ is the set of all possible q-grams over some alphabet.

We start with a description of BF.

- Setup takes as input the length ℓ of the BF encoding and the number $\boldsymbol{k}$ of hash functions, both of which are positive integers. Next, $\boldsymbol{k}$ hash functions $\mathsf{Hash}_i : \mathcal{R} \to \{1, \ldots, \ell\}$ are randomly sampled. Then, for each $g \in \mathcal{R}$, its encoding $E(g) \in \mathbb{F}_2^\ell$ is generated as follows:

$$E(g)[i] = \begin{cases} 1, & \exists j \in \{1, \ldots, \boldsymbol{k}\} : \mathsf{Hash}_j(g) = i \\ 0, & \text{else} \end{cases} \tag{5}$$

  The outputs are $\Pi_{sec} = \{E(g) | g \in \mathcal{R}\}$ and $\Pi_{pub} = \Pi_{in}$.
- Enc takes Π_{sec} and a record R as inputs and outputs the encoding $E(R)$ of the record which is defined as follows:

$$E(R) = \bigvee_{g \in R} E(g). \tag{6}$$

  Here, $\bigvee$ is the bitwise OR-operation, applied to each index i separately. That is, $E(R)[i] = \bigvee_{g \in R} E(g)[i]$ for each $i \in \{1, \ldots, \ell\}$.

We stress that the focus of this description is to illustrate the structural properties, i. e. the vectors $E(g)$ serve as a kind of basis that are combined by the $\vee$ operation to get the encoding $E(R)$ of the full record. In practice, it is not necessary to precompute and share all vectors $E(g)$. Instead, one could use salted hash functions and share the secret salts as part of Π_{sec}. The encodings $E(g)$ would then be computed on the fly when needed.

Moreover, we omitted a description of Link as different choices do exist, e. g. , the Jaccard or Dice coefficients or the Hamming weight, all using the same Enc procedure. As only the latter is relevant for the attack, a discussion on the instantiation of Link is out of scope.

The BAD scheme is almost identical to BF with the following differences. The Setup procedure takes as additional parameters two positive integers ℓ' and t and randomly samples a matrix $M \in \mathbb{F}_2^{\ell' \times \ell}$ where the Hamming weight of each row is bounded by t. In the Enc procedure, the computation of the final encoding Eq. (6) is replaced by

$$E(R) = M \times \left(\bigvee_{g \in R} E(g) \right). \tag{7}$$

The additional matrix multiplication was introduced by [1] to realize some diffusion in the encoding process. Indeed, they showed that while GMA breaks BF, it is ineffective against BAD. At the time of writing, BAD is the only secure non-interactive PPRL scheme. However, to maintain a sufficient level of matching quality, the authors suggested rather small values for t and $\boldsymbol{k}$, e. g. $t = 10$ and $\boldsymbol{k} = 10$ for BAD encodings of length 1000. The result is that the encodings $E(R)$ are rather sparse if $|R|$ is very low, which will be exploited by our new attack.

3 Security Model

To evaluate the security of a PPRL scheme, the commonly adopted attacker model is an attacker who has access to the public parameters Π_{pub}, encoded records and potentially also plain records. The attacker then aims to re-identify the individuals represented by the encoded records or at least to extract some partial information about these records.

So far, research on PPRL still lacks a concise security definition. In the following, we propose and discuss such a definition and address afterwards how it relates to existing attacks. For the remainder of this section, let $\mathcal{S} = (\mathcal{S}.\mathsf{Setup}, \mathcal{S}.\mathsf{Enc}, \mathcal{S}.\mathsf{Link})$ be a PPRL scheme with respect to $\mathsf{Link}_{\text{plain}}$ and Π_{in} be its input parameters. The proposed security definition is based on the concept of similarity graphs that we define as follows:

Definition 2 (Similarity Graph). *Let D be some database with records $R_1, \ldots, R_\ell$. We construct a graph $G = (N, E)$ with the nodes $N = \{n_1, \ldots, n_\ell\}$ such that there is a $1:1$-mapping between nodes n_i and records R_i and*

$$(n_i, n_j) \in E \Leftrightarrow \mathsf{Link}_{\text{plain}}(R_i, R_j) = 1. \tag{8}$$

The similarity graph $G(D)$ (with respect to $\mathsf{Link}_{\text{plain}}$) is defined as the isomorphism class of G.

Note that learning the similarity graph $G(D)$, given an encoding of D, is the core functionality of a PPRL scheme. Consequently, an intuitive approach for defining security is to require that an attacker cannot learn anything besides the similarity graph. However, this may be too strict in general. To make the definition more flexible, we also include the notion of a leakage function $\mathcal{L}$ as, for example, is common for symmetric searchable encryption. An example for $\mathcal{L}$ will be given below. To formally capture the notion of security, we use the standard approach of an indistinguishability game.

Definition 3 (Security Definition). *The security game with respect to $\mathcal{S}$ and a leakage function $\mathcal{L}$ is played between an attacker $\mathcal{A}$ and an oracle $\mathcal{O}$.*

1. *At the beginning, the oracle $\mathcal{O}$ runs $\mathcal{S}.\mathsf{Setup}(\Pi_{in})$ to get the secret parameters Π_{sec} and public parameters Π_{pub}. The latter are handed to $\mathcal{A}$.*
2. *$\mathcal{A}$ now chooses two databases $D_0 \neq D_1$ with $G(D_0) = G(D_1)$ and $\mathcal{L}(D_0) = \mathcal{L}(D_1)$ and hands these two databases to the oracle $\mathcal{O}$.*
3. *The oracle now uniformly samples a bit $b \in \{0, 1\}$ and executes $\mathcal{S}.\mathsf{Enc}(\Pi_{sec}, .)$ to get the encoded database $E(D_b)$ from D_b. The encoded database is given to $\mathcal{A}$.*
4. *The attacker $\mathcal{A}$ outputs a bit b' and wins the game if $b = b'$.*

The advantage $Adv(\mathcal{A})$ of $\mathcal{A}$ is defined as $\left| Pr[b = b'] - \frac{1}{2} \right|$.

The scheme $\mathcal{S}$ is called $(\mathcal{L}, \varepsilon)$-secure if for any probabilistic polynomial time attacker $\mathcal{A}$ (polynomial in the size of Π_{sec}) it holds that $Adv(\mathcal{A}) \leq \varepsilon$.

All existing attacks against PPRL schemes, including those mentioned in Section 1 and our new H-GMA (Section 4), aim for identifying structural properties of some database D, given its encoding $E(D)$, where these properties go beyond what is revealed by the similarity graph, e.g., that D contains a particular record. Thus, the applicability of such an attack directly implies the insecurity of the scheme according to Definition 3 as long as the observable property is not covered by the leakage function $\mathcal{L}$: An attacker chooses two databases D_0 and D_1 such that the observable property is given in only one of these two, e. g. D_0. Then, given $E(D_b)$ the attacker runs this attack to check if the property does hold for D_b. If so, it outputs $b' = 0$, otherwise $b' = 1$. Intuitively, the more effective the attack is in identifying the respective property, the higher the advantage of

the attacker to win the game. This shows that resistance against these attacks is necessary for security.

With respect to evaluating the security of PPRL schemes, note that none of the existing schemes, including the newly proposed SAUL scheme (Section 5), achieve the highest level of security, i. e. , $\mathcal{L} \equiv \perp$. The reason is that it holds for all these schemes that the encodings of very similar records leak *how* similar they are to each other (cf. Fig. 14 for the case of SAUL).

Therefore, we consider a (slightly) relaxed security definition by taking into account a leakage function $\mathcal{L}_{sim}$ that outputs for each edge of the similarity graph $(n_i, n_j) \in E$, that is, similar records, the corresponding similarity score $\mathsf{sim}_{plain}(R_i, R_j)$ (cf. Section 2.2). While this leakage goes beyond what is revealed by the similarity graph, we claim that it may be tolerable in practice: Assuming a record length of 30 bigrams, a Dice similarity of 0.5 on the plaintext will lead to an agreement probability of around 0.503 on the encodings when using BAD with $\boldsymbol{k} = 10$ and $t = 10$ and 0.506 when using SAUL with $\boldsymbol{k} = 4$ (cf. Eq. (34)). This only marginally differs from the agreement probability of 0.5 we would expect for random bitstrings. As can be seen from Fig. 1, the share of record pairs with a Dice similarity ≥ 0.5 is less than 0.5% for typical databases. Thus,

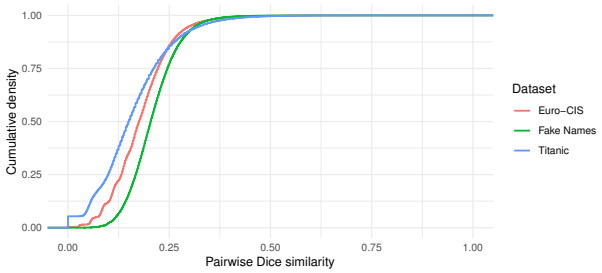

Figure 1: Cumulative distribution function of pairwise Dice similarities in different datasets.

only a very small fraction of records is actually affected by the information leak. Still, it is an open question whether information leakage is avoidable at all.

4 Breaking BAD

In this section, we present the first attack against BAD. To this end, we first identify two particular properties of BF that also extend to BAD in Section 4.1. Then, we describe an adaptation of the GMA that exploits these properties in Section 4.2. Finally, in Section 4.3, we implement the attack and run it against a simulated database of personal information.

4.1 Properties of BF and BAD

4.1.1 Hamming Weight Correlation. The first property we show is a strong correlation between the size $|R|$ of a plaintext record R and the Hamming weight $\mathrm{HW}(E(R))$ of the corresponding encoded record. Fix some q-gram $g \in \mathcal{R}$ and index $i \in \{1, \ldots, \ell\}$. By construction (cf. Eq. (5)), it holds that $E(g)[i] = 0$ if and only if $\mathrm{Hash}_j(g) \neq i$ for all $j \in \{1, \ldots, \ell\}$. It follows

$$p_0(E(g)[i]) = \left(\frac{\ell - 1}{\ell}\right)^{\boldsymbol{k}} \tag{9}$$

where the probability is taken over the random choices made in Setup. Because of Eq. (6), it holds for any record R

$$p_0(E(R)[i]) = \prod_{g \in R} p_0(E(g)[i]) = \left(\frac{\ell - 1}{\ell}\right)^{|R| \cdot \boldsymbol{k}} \tag{10}$$

Fig. 2 shows the value Eq. (10) for different sizes $|R|$. As one can see, it is more likely that a bit is zero unless $|R|$ is very high.

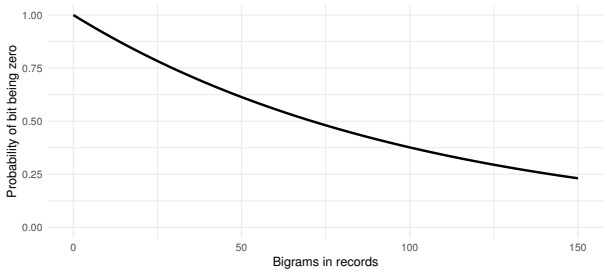

Figure 2: Estimated probability of a BF bit being zero (Eq. (10)) for different numbers of included q-grams, $\boldsymbol{k} = 10, \ell = 1024$.

If we assume that these probabilities hold for all i independently at the same time[2], the expected Hamming weight $\mathrm{HW}(E(R))$ is

$$\mathbb{E}(\mathrm{HW}(E(R))) = \left(1 - \left(\frac{\ell - 1}{\ell}\right)^{|R| \cdot \boldsymbol{k}}\right) \cdot \ell. \tag{11}$$

The expected Hamming weight $\mathbb{E}(\mathrm{HW}(E(R)))$ is strictly increasing with $|R|$. That is, if $|R|$ is low, $\mathrm{HW}(E(R))$ is expected to be close to 0. However, this value tends to ℓ with increasing $|R|$. This relationship is clearly visible in Fig. 3, which shows the experimentally determined Hamming weights of BF encoded records of varying length. The black line indicates the expected Hamming weight per Eq. (11).

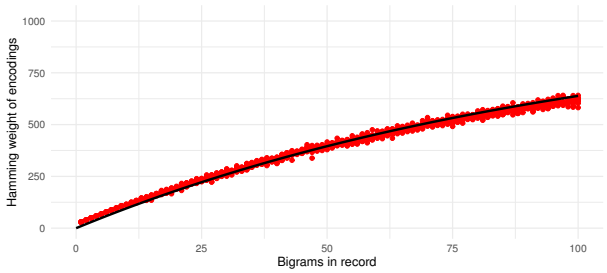

Figure 3: Observed Hamming weight of BF encoded records of varying length, $\boldsymbol{k} = 10, \ell = 1024$. Estimated Hamming weight in black.

These results can be transferred to the case of BAD in a straightforward manner through the application of the piling-up lemma [9]. While a low $|R|$ still implies small Hamming weight $\mathrm{HW}(E(R))$, the weight tends to $\ell/2$ for increasing $|R|$. Again, this phenomenon can be clearly seen in Fig. 4, which is a plot of the empirically observed Hamming weight of BAD encoded records of varying length. The expected Hamming weight is shown in black.

[2]As the total number of hash functions is bounded, this is not quite correct. However, experiments confirmed that this is a good approximation.

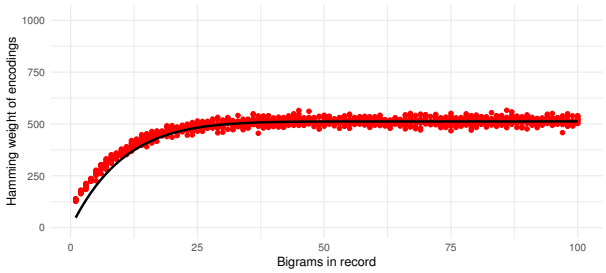

Figure 4: Observed Hamming weight of BAD encoded records of varying length, $k = 10, \ell = \ell' = 1024, t = 5$. Estimated Hamming weight in black.

4.1.2 Linear Approximability. The property from Section 4.1.1 can be used to match low weight encoded records $E(R)$ to plaintext records of small size. In practice, however, such events rarely happen. Specifically, records naturally do not have a very small size $|R|$. In the following, we argue that it is possible to combine several (encoded) records so that the results are "records" of low weight. This fact will later be exploited in our new attack.

In a nutshell, the idea is to linearly approximate the OR-operation $\bigvee$ in Eq. (6) by XOR. This is motivated by the fact that for any two binary values x_1 and x_2, it holds that $x_1 \vee x_2 = x_1 \oplus x_2$ unless $x_1 = x_2 = 1$. If we assume that both variables have the same probability p_1 to be one, it follows that

$$Pr\left[x_1 \vee x_2 = x_1 \oplus x_2\right] = 1 - p_1^2. \tag{12}$$

Note that the lower the probability p_1, the higher the probability that the linear approximation is correct. For the parameter choices suggested in [1], it actually holds that $p_1(E(g)[i])$ is very low for any index i and q-gram g as we have shown in the previous subsection. Given this, it intuitively holds

$$\begin{aligned}\bigoplus_{j=1}^{m} E(R_i) &= \bigoplus_{j=1}^{m} \bigvee_{g\in R_j} E(g) \approx \bigoplus_{j=1}^{m} \bigoplus_{g\in R_j} E(g) \\ &= \bigoplus_{g\in\Delta(R_1,\ldots,R_m)} E(g) \approx E(\Delta(R_1,\ldots,R_m)).\end{aligned}$$

Note that if $\Delta(R_1,\ldots,R_m) \approx \emptyset$, it follows from the property described in Section 4.1.1 that $E(\Delta(R_1,\ldots,R_m)) \approx \mathbf{0}$. This will be used in our new attack for matching selections of encoded records to selections of plaintext records and eventually lead to the re-identification of single records.

Next, we are interested in a lower bound for the probability that the following equation does hold:

$$\bigoplus_{j=1}^{m} \bigvee_{g\in R_j} E(g)[i] = \bigvee_{g\in\Delta(R_1,\ldots,R_m)} E(g)[i], \tag{13}$$

where $i \in \{1,\ldots,\ell\}$ is an arbitrary index and $R_1,\ldots,R_m$ is an arbitrary selection of records. Let $\Delta = \Delta(R_1,\ldots,R_m)$. We consider now two cases:

(1) It holds for *all* q-grams $g \in \bigcup_{j=1}^{m} R_j$ that $E(g)[i] = 0$. In that case, Eq. (13) trivially holds.
(2) There is exactly *one* q-gram $g^* \in \bigcup_{j=1}^{m} R_j$ with $E(g^*)[i] = 1$ while $E(g)[i] = 0$ for all remaining $g \neq g^*$. If $g^* \in \Delta$, then the right hand side of Eq. (13) is equal to 1, as exactly one of the addends is 1. By definition of Δ, the number of indices j such that $g^* \in R_j$ is odd. Hence, the left hand side of Eq. (13) is = 1 as well and the equality holds. With similar arguments, one can show that in the case of $g^* \notin \Delta$, both sides of Eq. (13) are equal to 0 and the equality holds as well.

It follows that a lower bound for the probability of Eq. (13) is the probability that at most one $E(g)[i]$ is equal to 1, which is

$$p_0^n + n \cdot p_0^{n-1} \cdot (1 - p_0) \tag{14}$$

with $n = \left|\bigcup_{j=1}^{m} R_j\right|$. With similar arguments as in Section 4.1.1, it follows that

$$\begin{aligned}&\mathbb{E}\left(\mathrm{HD}\left(\bigoplus_{j=1}^{m} \bigvee_{g\in R_j} E(g), \bigvee_{g\in\Delta(R_1,\ldots,R_m)} E(g)\right)\right) \\ \leq\ &\left(1 - p_0^n + n \cdot p_0^{n-1} \cdot p_1\right) \cdot \ell.\end{aligned}$$

4.1.3 Extension to BAD. The two properties identified for BF (Section 4.1.1 and Section 4.1.2) naturally extend to BAD due to the fact that the matrix M is sparse.

This means that the probability for a single encoding bit to be zero (cf. Eq. (10)) decreases with t. In consequence, the probability given in Eq. (14) for the equality specified in Eq. (13) decreases as well with t. More precisely, let $\varepsilon := p_0^n + n \cdot p_0^{n-1} \cdot (1 - p_0) - 1/2$. Then, according to the piling-up lemma [9], the probability that Eq. (13) holds in the case of BAD is $\frac{1}{2} + 2^{t-1} \cdot \varepsilon^t$. Fig. 5 displays this probability for different choices of t and record sizes. One can see that the linear approximation still is more likely to hold, even if both values are large. Our experiments confirm (Section 4.3) that this is still sufficient for a successful attack.

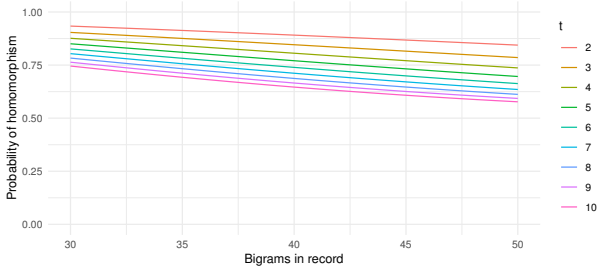

Figure 5: Lower bound for the probability that the linear approximation holds for a bit in BAD encodings for various record sizes and diffusion parameters, $k = 10, \ell = \ell' = 1024$.

4.2 Attack Description

In the following, we introduce homomorphism-based graph matching attacks (H-GMAs), a novel attack exploiting the approximate homomorphism of BAD encodings. Our attack follows the same attacker model as standard GMAs by assuming an honest-but-curious attacker, who has access to a database of BAD encoded records, $E(D)$, and an auxiliary plaintext database D^*. The plaintext database is assumed to overlap with the encoded database in terms of attributes and individuals included [16]. All other information, particularly the secrets Π_{sec}, remain unknown to the attacker.

In the attack, we search for structural relationships among the records, represent these in two separate graphs for the plaintext and the encoded data, and then apply a standard GMA methodology for record re-identification. In that sense, our attack shares some similarities with the GMA of [16], while it strongly differs in how similarities between encoded and plain records are identified. Overall, the H-GMA follows a three step approach:

Step 1: Discovery of Structural Relationships As established before, the property of BAD described in Section 4.1.1 may be used for a re-identification attack by linking plaintext records $R \in D^*$ with $|R|$ small to encodings $E(R') \in E(D)$ with $\mathrm{HW}(E(R'))$ low. However, the size of records in a database typically does not vary sufficiently to make such an attack useful in practice. Moreover, records of very small size are not to be found. Instead, the H-GMA exploits that the linear combination $\bigoplus_{j=1}^{m} \bigvee_{g \in R_j} E(g)$ is a good approximation of $\bigvee_{g \in \Delta(R_1,\ldots,R_m)} E(g)$ (Section 4.1.2). Note that if $|\Delta(R_1, \ldots, R_m)|$ is small, it is expected that $E(\Delta(R_1, \ldots, R_m))$ has a small Hamming weight.

More concretely, the attacker searches for tuples of records $(R_1, \ldots, R_m)$ where $2 \leq m \leq m_{max}$ whose combined symmetric difference is close to the empty set:

$$\Delta(R_1, \ldots, R_m) \approx \emptyset \tag{15}$$

The condition is satisfied if the cardinality of the resulting symmetric difference is below a pre-defined threshold, τ_p. This accounts for minor differences in the plaintext data, such as spelling variations. Let the set of all such identified plaintext tuples be denoted by $\mathcal{Q}_p$.

Next, the attacker performs an analogous search in the encoded database $E(D)$. The attacker searches for tuples of encoded records such that their combined XOR sum approximates the zero-vector:

$$\bigoplus (E(R_1), \ldots, E(R_m)) \approx \mathbf{0} \tag{16}$$

Here, the approximate equality is resolved by computing the Hamming weight of the resulting vector. If the Hamming weight is below the threshold τ_e, the tuple is considered a match. Let the set of all such identified encoded tuples be denoted by $\mathcal{Q}_e$.

Step 2: Construction of Relationship Graphs The sets of tuples discovered in the previous step are used to construct two graphs: a plaintext graph $G_p = (N_p, E_p)$ and an encoded graph $G_e = (N_e, E_e)$. The nodes of the graphs, N_p and N_e, are the individual records in D^* and $E(D)$, respectively. The edges are defined by the identified tuple sets:

- For every tuple $(R_1, \ldots, R_m) \in \mathcal{Q}_p$, the set of undirected edges $\{(R_i, R_j) \mid 1 \leq i < j \leq m\}$ is added to the edge set E_p.
- Similarly, for every tuple $(E(R)_1, \ldots, E(R)_m) \in \mathcal{Q}_e$, the corresponding clique of edges is added to the edge set E_e.

This forms a clique between the records belonging to the same tuple. Due to the discussed properties of BAD, the resulting graphs are expected to share isomorphic subgraphs. In particular, the neighborhood of a node, i. e. the number and structure of the cliques it belongs to, are expected to be highly similar in G_p and G_e. This can be exploited to identify these subgraphs and their respective node correspondences, enabling the attacker to link encoded records to their plaintext counterparts. This is the primary goal of the next step.

Step 3: Graph Matching and Re-Identification With the two relationship graphs constructed, the final phase of the attack proceeds as a standard GMA, employing the approach of [16]. The objective is to find a mapping between the nodes of G_p and G_e. This process consists of three sub-steps:

(1) **Node Embedding:** The embedding algorithm *Node2Vec* [6] is applied to G_p and G_e. This produces low-dimensional vector representations (embeddings) of each node, capturing the properties of a node's neighborhood.
(2) **Embedding Alignment:** Since the two sets of embeddings are in different vector spaces, they have to be aligned using Wasserstein Procrustes, producing a transformation matrix that maps the encoded embedding space onto the plaintext embedding space.
(3) **Bipartite Matching:** After applying the transformation to the encoded embeddings, the attacker computes the pairwise cosine distances between the set of aligned encoded embeddings and the set of plaintext embeddings. A bipartite minimum weight matching is then used to find the most likely correspondences.

This final mapping provides a direct re-identification of the records in the encoded database $E(D)$ with their counterparts in the plaintext database D^*.

4.3 Attack Evaluation

We experimentally evaluated the efficacy of H-GMAs against the BAD encoding scheme. Since the H-GMA relies on the presence of structural relationships in the plaintext dataset, standard PPRL datasets such as *Titanic*, *Fake Names* or *Euro-CIS* could not be used (See Section 5.4.2 for details on these datasets). Instead, the attack was run on a synthetic dataset of 1000 unique records. To create this dataset, 40 given names, 25 surnames, and 2 cities were randomly selected from the US-English dataset of the *Fake Name Generator*[3], and 1000 unique combinations of these values were sampled. By restricting the number of possible values the attributes could have we ensure that combinations of records exist for which the symmetric difference is almost empty. In accordance with [1, 16], an encoding length of 1024 was chosen. On the plaintext, the threshold τ_p was set to 5, whereas the encoded threshold τ_e was dynamically adjusted in each run so that the number of identified tuples on the encoded data did not exceed the number of tuples found on the plaintext by more than 10% .

As can be seen from Fig. 6, which illustrates the attack's success rate, i.e., the share of correctly re-identified records, the attack is highly effective across a wide range of parameters. Depending on the exact encoding settings, our attack is capable of re-identifying up to 91.6% of records (for $\boldsymbol{k} = 5, t = 6$). Crucially, the H-GMA is effective even against parameter settings previously considered secure: For BADs of length 1000, the authors of [1] recommend $t = 10$ and $\boldsymbol{k} = 10$. In this setting, our attack correctly re-identifies 905 out of 1000 records.

Overall, the results align well with the approximate homomorphism of BAD discussed in Section 4.1: A larger $\boldsymbol{k}$ increases the density, i. e. , the number of 1-bits of the intermediate BF (see Eq. (10)). This in turn increases the probability of a given bit being set by more than one q-gram. Overall, this degrades the homomorphic property of the BF layer by increasing the probability that a given

[3] https://www.fakenamegenerator.com

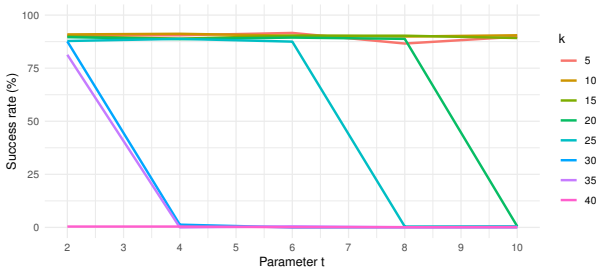

Figure 6: Success rate of the H-GMA against BAD encoded records for various values of k and t.

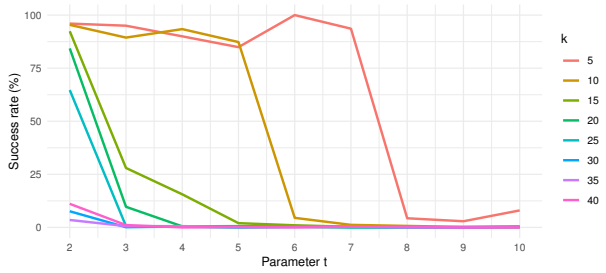

Figure 7: Success rate of a standard GMA against BAD encoded records for various numbers of hash functions (k) and diffusion parameters (t).

bit represents more than one q-gram (see Eq. (14)). This degradation introduces noise into the structural relationships that the H-GMA aims to exploit, thereby reducing its efficacy.

Notably, the influence of t is less substantial for smaller values of $\boldsymbol{k}$ (e.g., $\boldsymbol{k} \leq 15$). For instance, the success rates only drop from 90.5% ($\boldsymbol{k} = 5, t = 2$) to 89.8% ($\boldsymbol{k} = 5, t = 10$) and from 90.0% ($\boldsymbol{k} = 15, t = 2$) to 89.2% ($\boldsymbol{k} = 15, t = 10$). This is also explained by our theoretical framework: The diffusion step aggregates the homomorphism from t intermediate bits. When $\boldsymbol{k}$ is small, the probability that all of those t represent at most one q-gram is high and so is the probability that Eq. (13) holds. However, when $\boldsymbol{k}$ is large, the probability specified in Eq. (14) is inherently lower. In this scenario, the piling-up lemma [9] dictates that increasing t leads to a rapidly diminishing chance that Eq. (13) holds. Once t surpasses a critical value, this probability is so close to 0.5 that the homomorphism is effectively lost.

To further highlight the increased performance of the H-GMA compared to standard GMAs, we attacked the same dataset that was used for evaluating the H-GMA with a standard GMA. The attack parameters are given in Table 1.

Table 1: Attack parameters for experimental evaluation of GMA and H-GMA.

Parameter	Value
Thresholding quantile	0.9
Embedding dimensionality	128
Context size	10
P	250
Q	300
Embedding training epochs	5
Regularization convex initialization	1
Regularization Wasserstein alignment	0.33
Learning rate	200

As can be seen from Fig. 7, the standard GMA was able to successfully re-identify up to 100% in certain scenarios ($\boldsymbol{k} = 5, t = 6$). However, success rates degraded drastically with increasing $\boldsymbol{k}$ and t. In particular, the attack only re-identified 2 out of 1000 records for the (supposedly secure) parameter choice of $\boldsymbol{k} = 10, t = 10$, which is substantially less than the 905 records correctly re-identified by the H-GMA.

We also measured the duration of our attack, as shown in Figure 8. The primary determinant for the attack's duration is the encoding parameter $\boldsymbol{k}$. This is to be expected, as larger values of $\boldsymbol{k}$ induce more noise in the data, leading to a slower convergence of the embedding alignment step. Nevertheless, the overall runtime remained below one hour, even for large values of $\boldsymbol{k}$.

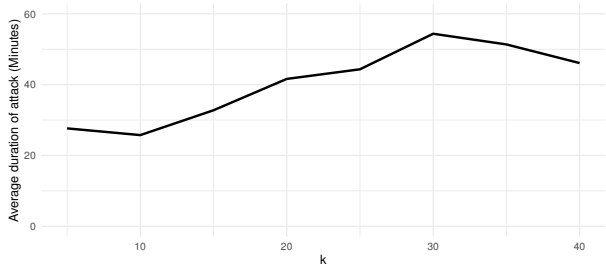

Figure 8: Average duration of the H-GMA against BAD encoded records for various values of k of hash functions, averaged over all diffusion parameters.

4.4 $(\mathcal{L}_{sim}, \varepsilon)$-Security of BAD

The success of the H-GMA implies that BAD fails to meet $(\mathcal{L}_{sim}, \varepsilon)$-security. To show this, we conducted an experiment mimicking the indistinguishability game described in Definition 3: Two artificial datasets D_0 and D_1 were generated and encoded with BAD and parameters $\boldsymbol{k} = 10$ and $t = 10$. Each database consisted of 1000 equally sized (30 bigrams), random and mutually dissimilar records. Since all pairwise similarities in both datasets were below the threshold for which similarities can be computed on BAD encoded data (see Fig. 9), their similarity graphs are identical, namely the empty graph. The key difference is that in database D_0, 500 records are linear combinations of other records from the database, whereas in D_1 there are no such structural relationships. In this setting, $\mathcal{A}$ was able to perfectly distinguish between D_0 and D_1 by using our H-GMA methodology.

5 Better Call SAUL

In this section we introduce a novel PPRL scheme, named Scheme for Anonymous Utility-preserving Linkage (SAUL). We show that

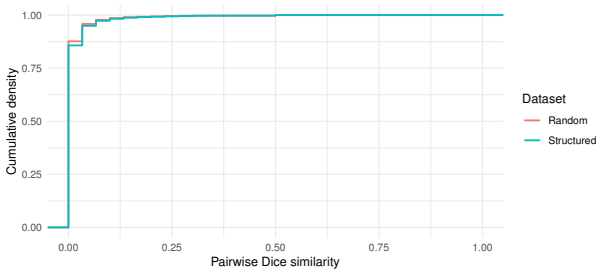

Figure 9: Cumulative distribution function of pairwise Dice similarities on the artificial datasets.

SAUL is not only secure against previous attacks but also against the one described in Section 4.2. At the same time, SAUL provides better linkage quality than BAD while being only marginally slower.

5.1 Design Rationale

The encoding of a record R shall be computed from the q-grams it contains, denoted as $g \in R$. Thus, as in the case of BF and BAD, we are looking for a function (family) f such that

$$E(R) = f(E(g_1), \ldots, E(g_n)) \tag{17}$$

where, for the sake of simplicity, we initially restrict the output of the encoding function to one bit. We can formulate the following three requirements that f should fulfill:

(1) All inputs shall have the same potential impact. As a consequence, f must be symmetric.
(2) To avoid correlation, all outputs should be possible. That is, f should be balanced and both bit values should have roughly the same probability.
(3) To avoid homomorphism-based attacks, it should not be possible to approximate f by functions of lower degree, in particular linear functions.

A straightforward choice for f is the majority function, as it fulfills the first two requirements. Additionally, it is known to exhibit optimal algebraic immunity, e. g. [2], meaning that it cannot be canceled out by functions of lower degree. It does, however, not have optimal nonlinearity. For this reason, we consider the sum of several majority function executions to reduce the probability that the linear approximation is correct (more on this in Section 5.3.2).

5.2 Description

Circling back to Definition 1, we describe SAUL in terms of the algorithms constituting a PPRL scheme:

- Setup takes as input two positive integers: ℓ, the length of the encoding, and $\boldsymbol{k}$, the number of intermediate representations per q-gram. Moreover, a real value τ_{enc} is given. For each possible q-gram $g \in \mathcal{R}$, the setup algorithm uniformly samples a total of $\boldsymbol{k}$ vectors in $\mathbb{F}_2^\ell$. These constitute a kind of intermediate encodings and hence are denoted by $E_1(g), \ldots, E_{\boldsymbol{k}}(g)$. The outputs of the setup phase are $\Pi_{sec} = \{E_i(g)\}_{g\in\mathcal{R}, i\in\{1,\ldots,\boldsymbol{k}\}}$ and $\Pi_{pub} = \{\tau_{enc}\}$.
- Enc takes Π_{sec} and a record R as inputs and outputs the encoding $E(R)$ of the record which is defined as follows. Let $R = \{g_1, \ldots, g_n\}$. Then, first $\boldsymbol{k}$ intermediate encodings of R are computed by

$$E_i(R) = \mathrm{maj}((E_i(g_1), \ldots, E_i(g_n))) \tag{18}$$

for $i = 1, \ldots, \boldsymbol{k}$. Then, the final encoding is generated by XORing these vectors:

$$E(R) = \bigoplus_{i=1}^{\boldsymbol{k}} E_i(R). \tag{19}$$

- Link takes as input two encoded records $E(R)$ and $E(R')$ and outputs 1 if and only if $\mathrm{HD}(E(R), E(R')) \leq \tau_{enc}$.

Again, this description of SAUL focuses on highlighting its structural properties. In practice, it will be more efficient to agree on a pseudo-random number generator (PRNG) algorithm and to distribute the seed as part of Π_{sec}, allowing all participants deterministically recreate the intermediate representations $E_j(g)$ of g.

5.3 Security

Following the approach of other works in the field, e. g. [1], we assess the security of SAUL by evaluating its resistance against the most relevant attacks known so far: pattern mining, GMA, and our new H-GMA.

5.3.1 Pattern Mining Attacks. The objective of *pattern mining attacks* [8] is to discover a reliable statistical correlation between the presence of specific q-grams in a plaintext record and a predictable pattern in its final encoding. Recall that

$$E(R)[i] = \bigoplus_{j=1}^{\boldsymbol{k}} E_j(R) = \bigoplus_{j=1}^{\boldsymbol{k}} \mathrm{maj}\Big(\big(E_j(g)[i]\big)_{g\in R}\Big) \tag{20}$$

for every index $i \in \{1, \ldots, \ell\}$ and record R. As the values $E_j(g)[i]$ are independently sampled for each i during setup, the only potentially observable pattern in the encodings are single bit values, i. e. whether $E(R)[i] = 0$ happens with a frequency that significantly deviates from $\frac{1}{2}$. More precisely, we investigate the question if certain sets of q-grams $\widetilde{R}$ can significantly influence the outcome of an encoding.

That is equivalent to the probability of $E(R)[i] \neq E\Big(R \cup \widetilde{R}\Big)[i]$ for some record R with $R \cap \widetilde{R} = \emptyset$. Again, due to the fact that the values $E_j(g)[i]$ are uniformly and independently sampled, this can be expressed by the probability $p(n, \widetilde{n})$ for integers $n, \widetilde{n}$ such that

$$p(n, \widetilde{n}) = Pr\left[E_j(R)[i] \neq E_j\Big(R \cup \widetilde{R}\Big)[i]\right] \tag{21}$$

with $n = |R|$ and $\widetilde{n} = \left|\widetilde{R}\right|$. Note that $E(R)[i] \neq E\Big(R \cup \widetilde{R}\Big)[i]$ if and only if $\left|\left\{j | E_j(R)[i] \neq E_j\Big(R \cup \widetilde{R}\Big)[i]\right\}\right|$ odd. Hence:

$$\begin{aligned} &Pr\left[E(R)[i] \neq E\Big(R \cup \widetilde{R}\Big)[i]\right] \\ &= \sum_{e=1,\ldots,\boldsymbol{k}:\ e \text{ odd}} p(n, \widetilde{n})^e \cdot (1 - p(n, \widetilde{n}))^{\boldsymbol{k}-e}. \end{aligned} \tag{22}$$

Next, we investigate the expression $p(n, \widetilde{n})$ (Eq. (21)). Inequality holds if either $\mathrm{maj}\Big(\big(E_j(g)[i]\big)_{g\in R}\Big) = 0$ but $\mathrm{maj}\Big(\big(E_j(g)[i]\big)_{g\in R\cup\widetilde{R}}\Big) = 1$ (case 0), or vice versa (case 1). Let $\mathrm{Zeroes}(E_j(R))$ be the number of all $g \in R$ with $E_j(g) = 0$ (and likewise $\mathrm{Zeroes}(E_j\Big(R \cup \widetilde{R}\Big)$, etc.)

Then case 0 translates to

$$\begin{aligned} \text{Zeroes}(E_j(R)[i]) &\geq \frac{n}{2} \text{ and} \quad (23)\\ \text{Zeroes}(E_j(R)[i]) + \text{Zeroes}(E_j(\widetilde{R})[i]) &< \frac{n}{2} + \frac{\widetilde{n}}{2}. \end{aligned}$$

The latter implies

$$0 \leq \text{Zeroes}(E_j(\widetilde{R})[i]) < \frac{n}{2} + \frac{\widetilde{n}}{2} - \text{Zeroes}(E_j(R)). \quad (24)$$

The probability that both Eq. (23) and Eq. (24) hold is

$$\sum_{a=\lceil \frac{n}{2} \rceil}^{n} \binom{n}{a} 2^{-n} \cdot \sum_{b=0}^{\lceil \frac{n+\widetilde{n}}{2} \rceil - a - 1} \binom{\widetilde{n}}{b} 2^{-\widetilde{n}} \quad (25)$$

Analogously, case 1 translates to

$$\begin{aligned} \text{Zeroes}(E_j(R)[i]) &< \frac{n}{2} \text{ and} \quad (26)\\ \text{Zeroes}(E_j(R)[i]) + \text{Zeroes}(E_j(\widetilde{R})[i]) &\geq \frac{n}{2} + \frac{\widetilde{n}}{2}. \end{aligned}$$

The latter implies

$$\widetilde{n} \geq \text{Zeroes}(E_j(\widetilde{R})[i]) \geq \frac{n}{2} + \frac{\widetilde{n}}{2} - \text{Zeroes}(E_j(R)). \quad (27)$$

The probability that both Eq. (26) and Eq. (27) hold is

$$\sum_{a=0}^{\lceil \frac{n}{2} \rceil - 1} \binom{n}{a} 2^{-n} \cdot \sum_{b=\lceil \frac{n+\widetilde{n}}{2} \rceil - a}^{\widetilde{n}} \binom{\widetilde{n}}{b} 2^{-\widetilde{n}} \quad (28)$$

Thus, $p(n, \widetilde{n})$ is given by the sum of Eq. (25) and Eq. (28). Figure 10 displays this probability for varying sizes of the plaintext records and patterns with $\widetilde{n} < n$. It shows that the probability of a bit flip is rather small, implying that the pattern does not significantly impact the encoding.

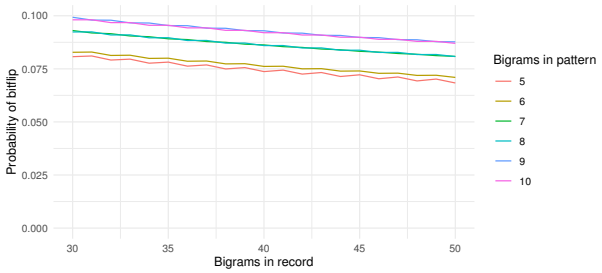

Figure 10: Probability of a bit flipping in SAUL encoded data due to the inclusion of a pattern in the plaintext. $k = 4, \ell = 1024$.

5.3.2 Homomorphism-based Graph Matching Attacks. As the necessity for a new scheme is motivated by our new attack H-GMA, it is natural to investigate the resistance of SAUL against this attack. Recall that H-GMA exploits the two properties identified in Section 4.1 for BF and BAD. We argue in the following that none of these hold for SAUL with a significant probability.

Hamming Weight Correlation. The first property is that records R of small size $|R|$ result in encodings of low Hamming weight. In the case of SAUL, the intermediate encodings $E_j(g)$ are uniformly sampled from $\mathbb{F}_2^\ell$. Thus, it holds for any index $i \in \{1, \ldots, \ell\}$ that $p_0(E_j(g)[i]) = p_1(E_j(g)[i]) = 1/2$. We consider $p_0(E_j(R)[i]) = \text{maj}\left((E_j(g)[i])_{g \in R}\right)$ for some record R. As the majority function is perfectly balanced for an odd number of inputs, it holds that $p_0(E_j(R)[i]) = 1/2$ for $|R|$ odd. It follows for the actual encoding that $p_0(E(R)[i]) = 1/2$ as well and hence the expected Hamming weight is $\ell/2$, irrespective of the concrete choice of R.

The case where $|R|$ is even is slightly different as the majority function has a higher chance to output zero. More precisely, it holds that

$$p_0\left(\text{maj}\left((E_j(g)[i])_{g \in R}\right)\right) = \frac{1}{2} + \underbrace{\frac{1}{2} \cdot \frac{\binom{n}{n/2}}{2^n}}_{=:\varepsilon(n)} \quad (29)$$

with $n = |R|$. Note that the term $\varepsilon(n)$ exponentially decreases with n. As can be seen in Fig. 11, the expected Hamming weight deviates from $\ell/2$ only for very small values of n.

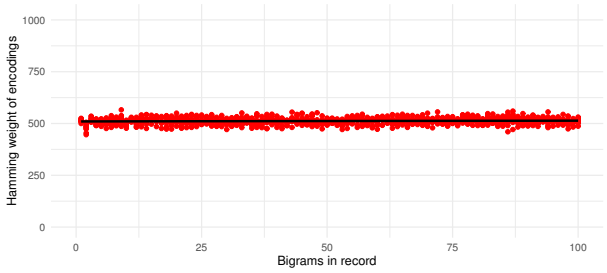

Figure 11: Observed Hamming weight of SAUL encoded records of varying plaintext length, $k = 4$, $\ell = 1024$. Averages in black.

Linear Approximability. It is known that the majority function does not provide optimal non-linearity. In fact, the best linear approximation of the majority function is given by

$$Pr\left[\text{maj}(x_1, \ldots, x_n) = x_i\right] = \frac{1}{2} + \underbrace{\frac{\binom{n-1}{(n-1)/2}}{2^{n+1}}}_{=:\varepsilon(n)}, \quad (30)$$

where x_i is any of the n inputs and n is odd. One can use this to approximate any of the intermediate encodings $E_j(R)$. Due to the piling-up lemma, this yields a linear approximation of the full record with probability

$$\frac{1}{2} + 2^{k-1} \cdot \varepsilon(n)^k = \frac{1}{2} + \frac{\binom{n-1}{(n-1)/2}^k}{2^{k \cdot n - 1}} \quad (31)$$

Thus, the probability that the linear approximation holds, tends to $1/2$ exponentially in k and n. Figure 12 plots Eq. (31) for various record sizes and values of k. Note that the Y-axis of the plot only covers the range from 0.5000 to 0.5015. It is clearly visible that, even for low values of k and short records, the probability of the linear approximation holding only marginally differs from $1/2$.

For $k > 4$, the probability is indistinguishable from 1/2 within machine precision. This suggests that SAUL exhibits no meaningful homomorphism that could be exploited in practice. This is in stark contrast to BAD, where the lower bound for the same probability ranged between 0.58 (50 bigrams, $t = 10$) and 0.93 (30 bigrams, $t = 2$).

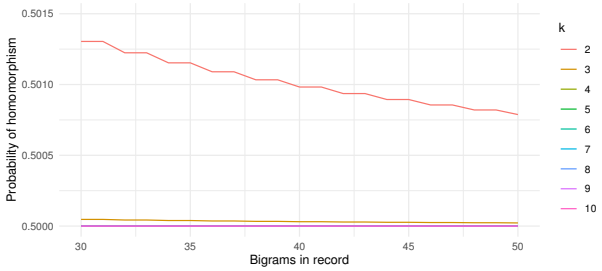

Figure 12: Probability that the linear approximation holds for a bit in SAUL encodings for various record sizes and parameters k, $\ell = 1024$.

Attack Evaluation. While SAUL is also subject to some correlation between record size and Hamming weight of the encoded records and also allows for some linear approximation, both do hold with significantly lower probability compared to BAD (or BF). Thus, it is to be expected that H-GMA will perform worse against SAUL than for BAD. We validated this by a series of experiments in which we re-ran the H-GMA described in Section 4.2 against a SAUL encoded version of the database. As can be seen from Fig. 13, which shows the success rate of the attack for varying parameters k, the H-GMA neither substantially, nor significantly outperforms random guessing (shown in red).

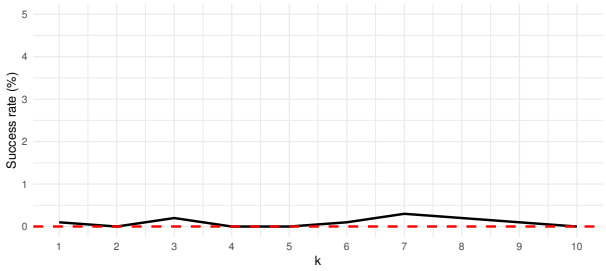

Figure 13: Success rate of the H-GMA against SAUL encoded records for various choices for k, $\ell = 1024$. Note the different scale of the Y-axis. Random baseline in red.

5.3.3 Graph Matching Attacks. As the purpose of PPRL schemes is to identify similar records from their encodings only, it is inevitable that for similar record pairs R and R', the corresponding encodings $E(R)$ and $E(R')$ leak the fact that they are similar. However, for several PPRL schemes it also holds that even if the records are not similar, some information is leaked from the encodings about how (dis-)similar the plaintext records are. In [16], it was postulated that security against GMA requires that the similarity of encoded records should always be about the same unless the plaintext records are similar. According to Eq. (34), this property holds for SAUL if parameter k is sufficiently large, which is also confirmed by experiments: Figure 14 shows the relationship between the Dice similarity of fixed-length plaintext record pairs and the Hamming similarity (ℓ minus Hamming distance) of their respective encodings.

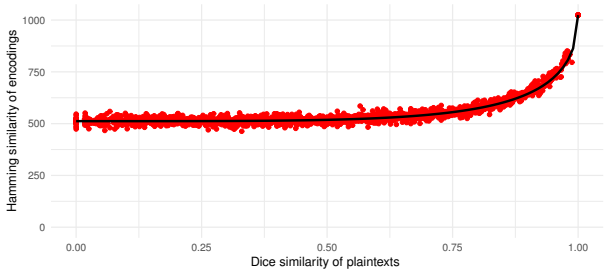

Figure 14: Relationship of Dice coefficients on plaintexts and Hamming similarity of SAUL encoded data. $k = 4$. Estimated Hamming similarity in black.

In addition, we did run a standard GMA against SAUL for different choices of k (see Fig. 15) and using the same dataset as for the H-GMA. As expected, the GMA is unable to outperform random guessing for $k \geq 3$.

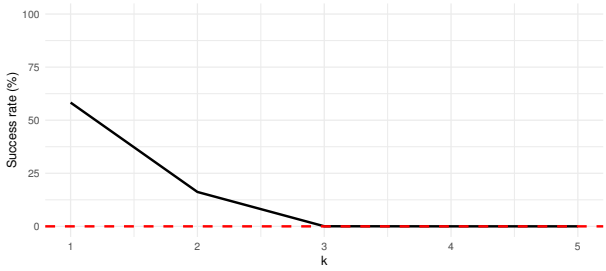

Figure 15: Success rate of the standard GMA against SAUL encoded records for various k, $\ell = 1024$. Random baseline in red.

5.3.4 $(\mathcal{L}_{sim}, \varepsilon)$-Security of SAUL. Our security analysis shows that none of the attacks known so far work against SAUL. More precisely, any additional q-gram adds another kind of random layer so that SAUL encodings exhibit properties of randomly chosen bitstrings where similarity is only observed if the underlying records are similar. This indicates that SAUL is secure with respect to Definition 3 and $\mathcal{L}_{sim}$. To further substantiate this claim, we replicated the indistinguishability game for databases encoded with SAUL and parameters $k = 4$ and $\ell = 1024$. Here, the attacker's success did not outperform random guessing. Of course, this does not represent a proof of security. Given that SAUL does not use any cryptographic building blocks and does not rely on any hardness assumptions, we do not see how one could reduce the security of SAUL to any other

assumption. However, an interesting question for future investigation is the overall design principle, namely to directly combine randomly chosen bitstrings such that the properties listed in Section 5.1 are met. For example, one could try to idealize the functions used in Eq. (18) and Eq. (19) and show that outputs are indistinguishable from random values, based on these idealized properties. We leave this as an open question for future research.

5.4 Linkage Quality

Besides providing security (cf. Section 5.3), it is likewise important that a PPRL ensures a high linkage quality. That is, for any similar pairs of records R and R', it should hold that this is correctly recognized by Link (see Section 5.2). In the case of SAUL, this means that the Hamming distance $\mathrm{HD}(E(R), E(R'))$ should be small. In the following, we first provide theoretical arguments for this property (Section 5.4.1) and support it afterwards by experiments (Section 5.4.2).

5.4.1 Theoretical Analysis. Let us consider two records, R and R'. For simplicity, we assume they have the same cardinality, $|R| = |R'| = n$. Let their Dice similarity be $s = \frac{2|R \cap R'|}{|R|+|R'|} = \frac{|R \cap R'|}{n}$. Observe that $s \in [0, 1]$ where larger values indicate a higher level of similarity. The following theorem allows to express the probability $Pr\left[E(R)[i] = E(R')[i]\right]$ for any index $i \in \{1, \ldots, \ell\}$ as a function in the similarity s.

Theorem 5.1. *It holds for any bit index i and any intermediate encoding index j that*

$$p_{agree} := Pr\left[E_j(R)[i] = E(R')j[i]\right] = \frac{1}{2} + \frac{1}{\pi}\arcsin(s) \tag{32}$$

Given this, it follows immediately with the help of the piling-up lemma [9] that

$$Pr\left[E(R)[i] = E(R')[i]\right] = \frac{1}{2} + 2^{k-1} \cdot \left(\frac{1}{\pi}\arcsin(s)\right)^k \tag{33}$$

$$= \frac{1}{2} + \frac{1}{2} \cdot \left(\frac{2}{\pi} \cdot \arcsin(s)\right)^k \tag{34}$$

Note that if the two records are similar, it holds that $s \approx 1$ and hence $\arcsin(g) \approx \frac{\pi}{2}$. Thus, the probability that the two bits agree is close to 1. However, with decreasing s, i. e. the records are less similar, the value $\arcsin(s)$ tends to 0 and hence the probability above tends to 1/2. This can also be seen in Fig. 14, where Eq. (34) is plotted as the black line. It remains to show Theorem 5.1.

Proof of Theorem 5.1. We show the following two claims:

- Claim 1: $p_{agree} = \frac{1}{2} + \frac{1}{\pi}\arcsin(\rho)$ for some value ρ
- Claim 2: $\rho = s$

It is obvious that the combination of these two claims imply the theorem.

Showing Claim 1. The generation of a bit

$$E_j(R)[i] = \mathrm{maj}\Big(\big(E_j(g)[i]\big)_{g \in R}\Big)$$

can be modeled as a Bernoulli process with $n = |R|$ trials, where the bit is 1 if and only if at least half of the trials produce a value of 1 (i. e. the majority of the bits $E_j(g)[i]$ is 1). Since the values were assigned to the q-grams uniformly and independently at random, the probability of each draw to be 1 is 0.5. Consequently, p_{agree} is equivalent to the probability that two such random processes yield the same majority outcome, given that they share $o = n \cdot s = |R \cap R'|$ common elements (whose outcomes are fixed).

To this end, we make use of the Central Limit Theorem which states that, for reasonably large n, the number of successes in a Bernoulli process converges to a normal distribution. Thus, the problem reduces to finding the probability that two correlated, normally distributed random variables (the number of trials yielding 1) have a value greater than the median of the distribution they were sampled from. Sheppard's theorem on median dichotomies [18] allows us to calculate this probability based on ρ, the Pearson correlation coefficient of the two samples, namely

$$p_{agree} = \frac{1}{2} + \frac{1}{\pi}\arcsin(\rho). \tag{35}$$

Showing Claim 2. To show that the Pearson correlation coefficient ρ is equal to the Dice similarity s, we decompose the sets R and R' into disjoint parts:

- $\mathcal{U} = R \setminus R'$: The unique q-grams in R. $|\mathcal{U}| = n - o$.
- $\mathcal{U}' = R' \setminus R$: The unique q-grams in R'. $|\mathcal{U}'| = n - o$.
- $\mathcal{I} = R \cap R'$: The common q-grams. $o = |\mathcal{I}|$.

To simplify the following presentation, we abbreviate $E_j(g)[i]$ to $E^*(g)$ and interpret the output of the latter as integers instead of elements of $\mathbb{F}_2$. We can now define two random variables, X and X', representing the sum of the q-gram bits for the two records:

$$X = \sum_{g \in \mathcal{U}} E^*(g) + \sum_{g \in \mathcal{I}} E^*(g) \tag{36}$$

$$X' = \sum_{g \in \mathcal{U}'} E^*(g) + \sum_{g \in \mathcal{I}} E^*(g) \tag{37}$$

Our goal is to compute the Pearson correlation coefficient of these two variables:

$$\rho(X, X') = \frac{\mathrm{Cov}(X, X')}{\sigma(X)\sigma(X')}, \tag{38}$$

where Cov is the covariance and σ is the standard deviation.

We start by determining the variance of the individual q-gram bits, which can be derived from the Bernoulli distribution with parameter $p = 0.5$:

$$\mathrm{Var}(E^*(g)) = p(1-p) = 0.5(1-0.5) = \frac{1}{4} \tag{39}$$

This allows us to calculate the variance of the overall sums. Recall that the variance of a sum of independent random variables (the q-gram bits) is the sum of their variances. Since X is the sum of n independent Bernoulli trials (composed of $n - o$ unique variables and o common variables), its variance is:

$$\mathrm{Var}(X) = n \cdot \mathrm{Var}(E^*(g)) = \frac{n}{4} \tag{40}$$

By symmetry, the variance of X' is identical:

$$\mathrm{Var}(X') = \frac{n}{4} \tag{41}$$

From this, the standard deviations are $\sigma(X) = \sigma(X') = \sqrt{n/4} = \frac{\sqrt{n}}{2}$.

Next, we calculate the covariance. Due to the bilinearity of covariance it holds that:

$$\begin{aligned}
&\mathrm{Cov}(X, X') \\
&= \mathrm{Cov}\Big(\sum_{g\in\mathcal{U}} E^*(g) + \sum_{g\in\mathcal{I}} E^*(g), \sum_{g\in\mathcal{U}'} E^*(g) + \sum_{g\in\mathcal{I}} E^*(g)\Big) \\
&= \mathrm{Cov}\Big(\sum_{g\in\mathcal{U}} E^*(g), \sum_{g\in\mathcal{U}'} E^*(g)\Big) + \mathrm{Cov}\Big(\sum_{g\in\mathcal{U}} E^*(g), \sum_{g\in\mathcal{I}} E^*(g)\Big) \\
&\quad + \mathrm{Cov}\Big(\sum_{g\in\mathcal{U}'} E^*(g), \sum_{g\in\mathcal{I}} E^*(g)\Big) + \mathrm{Cov}\Big(\sum_{g\in\mathcal{I}} E^*(g), \sum_{g\in\mathcal{I}} E^*(g)\Big)
\end{aligned} \tag{42}$$

Note that the sets $\mathcal{U}$, $\mathcal{U}'$ and $\mathcal{I}$ are mutually disjoint and their corresponding q-gram draws are independent. As the covariance between sums of independent variables is zero, Eq. (42) reduces to a single term, namely the covariance of a random variable with itself. This is, by definition, the variance of the variable:

$$\begin{aligned}
\mathrm{Cov}(X, X') &= \mathrm{Cov}\Big(\sum_{g\in\mathcal{I}} E^*(g), \sum_{g\in\mathcal{I}} E^*(g)\Big) \\
&= \mathrm{Var}\left(\sum_{g\in\mathcal{I}} E^*(g)\right) \\
&= o \cdot \mathrm{Var}(E^*(g)) = \frac{o}{4}.
\end{aligned} \tag{43}$$

By substituting the variance and covariance in Eq. (38) with our calculations from Eq. (41) and Eq. (43) we get:

$$\rho(X, X') = \frac{\mathrm{Cov}(X, X')}{\sigma(X)\sigma(X')} = \frac{o/4}{(\sqrt{n}/2)\cdot(\sqrt{n}/2)} = \frac{o/4}{n/4} = \frac{o}{n}. \tag{44}$$

Given the definition of the Dice coefficient it holds that:

$$s = \mathrm{Dice}(R, R') = \frac{2|R \cap R'|}{|R| + |R'|} = \frac{2o}{n+n} = \frac{o}{n} = \rho(X, X'). \tag{45}$$

This concludes the proof. □

Recall that we have restricted the analysis to the case that both records are of the same size. If this assumption does not hold, the simplification in Eq. (44) is no longer applicable and the correlation becomes

$$\rho(X, X') = \frac{\mathrm{Cov}(X, X')}{\sigma(X)\sigma(X')} = \frac{o/4}{(\sqrt{n}/2)\cdot(\sqrt{n'}/2)} = \frac{o}{\sqrt{n\cdot n'}}, \tag{46}$$

as n and n' are always non-negative. This can be interpreted as the ratio of the overlap to the geometric mean of the record sizes. While this is no longer equivalent to the Dice coefficient, it is still monotonically increasing with o for fixed n and n'.

5.4.2 Experimental Evaluation. We experimentally evaluated the linkage quality of SAUL in comparison to BAD and standard BF. The F_1-Score, which is defined as the harmonic mean of Precision and Recall, is used as the primary evaluation metric. In all experiments, symmetric matching based on the Dice similarity was deployed: A pair of records was considered a match if and only if its Dice coefficient exceeded 0.85 and each record was the mutually most similar counterpart to the other. Records for which no such counterpart existed remained unmatched. We used parameter settings of $\boldsymbol{k} = 20$ and $t = 10$ for BF and BAD, which was the lowest value of $\boldsymbol{k}$ for which the H-GMA did not succeed, and $\boldsymbol{k} = 4$ for SAUL. The choice of $\boldsymbol{k} = 4$ for SAUL was motivated by the fact that it was the smallest value of $\boldsymbol{k}$ that could be considered secure during the security analysis. For BF and BAD the implementations of [16] were used[4], whereas SAUL was implemented by ourselves from scratch based on NumPy. The resulting code was then parallelized and optimized using Numba. All experiments were conducted in a virtual machine with access to 20 cores of an AMD Epyc 9254 CPU and running Ubuntu 24.04.2 LTS.

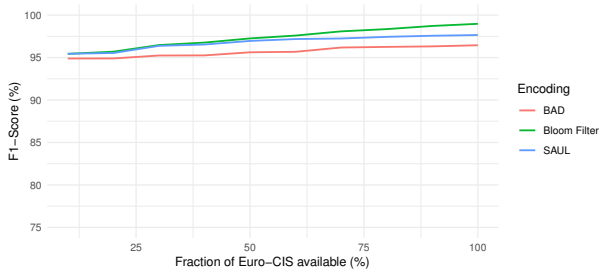

Figure 16: F_1-Score achieved by different encoding schemes when matching the Euro-CIS dataset. Symmetric matching based on Dice.

Datasets: The evaluation was conducted on two different types of datasets: First, we used the *Euro* dataset, which was created by the EU statistics office *Eurostat* specifically for evaluating record linkage schemes [5]. It contains three databases simulating data from a census, a customer information system (CIS) and a patient registry, as well as ground truth data. Specifically, we matched the 24,613 records of the Euro-CIS dataset to the 26,625 ground truth records based on name, surname, birthdate, gender, address and ZIP code. 1,841 of these attributes were missing in the CIS database and 16,100 were erroneous. To simulate real-world scenarios where databases might not perfectly align, we also created subsets by randomly sampling a certain fraction of records from the Euro-CIS dataset and matching these subsets against the full ground truth database.

To rule out the possibility of the linkage quality being skewed by characteristics of the dataset, a second type of data was obtained from the *Fake Name Generator*[6]. Here, we generated three datasets using the English (US), French, and German name sets. The results presented are the averages over these three datasets. Each dataset contained 100,000 records comprising a given name, surname, address, city, state, ZIP code, and birthday. Prior to matching, errors were introduced in a subset of records to test the robustness of the schemes against data quality issues: For each modified record, one attribute was altered by either truncation, dropping the attribute, or swapping two characters.

Performance with Databases of Varying Size: Figure 16 depicts the F_1-Scores achieved on the various samples of the Euro-CIS dataset. BF encoding provides the best results across all subsets of the database with F_1-Scores of up to 99%, highlighting the inherent trade-off between security and linkage quality. The F_1 scores obtained by matching BAD encoded data are the lowest throughout the experiments, ranging from 94.9% at the 10% sample to 96.4%

[4] https://github.com/SchaeferJ/graphMatching
[5] https://ec.europa.eu/eurostat/cros/content/job-training
[6] https://www.fakenamegenerator.com

when using the full database. This is consistently 1 to 2 percentage points below the BF results.

The experiments also show that SAUL achieves F_1-Scores that are around 0.5 to 1.5 percentage points higher than those of BAD across all tested samples and, for small database subsets, comparable to those of BF. For instance, when running on the 10% sample, SAUL achieves an F_1-Score of 95.4% (+0.5 pp vs. BAD, identical to BF). At the 50% sample, SAUL's score is 97% (+1.4 pp vs. BAD, −0.3 pp vs. BF) and when using the full database it is 97.7% (+1.3 pp vs. BAD, −1.3 pp vs. BF).

Performance with Varying Error Rates: To evaluate the influence of data errors on the linkage quality, we matched erroneous copies of the Fake Name datasets to error-free versions. The observed F_1-Scores for linkage are shown in Fig. 17.

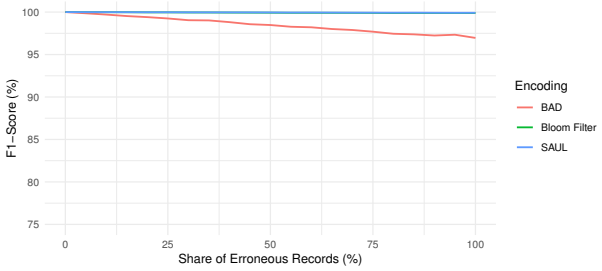

Figure 17: F_1-Score achieved by different encoding schemes when matching erroneous versions of the Fake Name datasets. Symmetric matching based on Dice.

Here, the difference in linkage quality between SAUL and BAD is even more pronounced: The F_1-Score for the BAD scheme degrades significantly as the share of erroneous records increases, dropping from 99.9% at 5% error rate to 97.9% with 70% erroneous records and further to 97% if all records contain errors. In contrast, the performance of both SAUL and the BF baseline remains stable, even in the presence of high error rates: For both schemes, the F_1-Score remains close to 99.99% as the error rates increase.

5.5 Scalability

To be practical for real-world applications, a PPRL encoding scheme must not only be secure and correct but also computationally efficient, especially when dealing with large datasets. We therefore evaluated the performance and scalability of SAUL and compared it to that of BADs and standard BFs. All schemes were implemented such that the encodings of the individual q-grams are pre-computed during setup using a seeded PRNG. Thus, encoding a record consists of retrieving the pre-computed encodings of the q-grams and combining them according to the respective algorithm.

Figure 18 is a visual representation of the benchmark results. All timings refer to the time it took to encode one of the databases during the evaluation of the linkage quality. As expected, standard Bloom filters generally provide the fastest encoding times, establishing a performance baseline. The inclusion of a diffusion layer in BAD causes a significant overhead, which translates into BAD requiring more than twice the time of BF when encoding the same database. Consequently, SAUL is the slowest of the three encoding

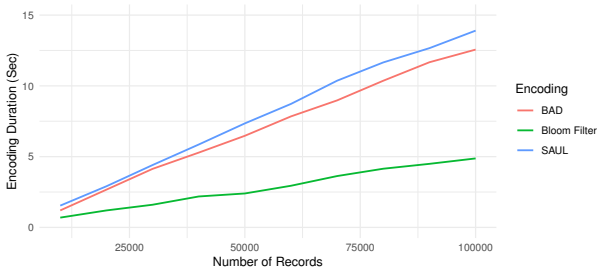

Figure 18: Comparison of encoding duration for Bloom filters, BADs, and SAUL on datasets of increasing size.

schemes. This can be attributed to the fact that computing a SAUL encoding involves even more computational steps, namely retrieving the intermediate representations of the q-grams, the majority calculations and the final XORs. For instance, encoding 100,000 records with SAUL took approximately 13.9 seconds, whereas the same task required about 12.6 seconds with BAD.

The evaluation also confirms that all three encoding schemes exhibit approximately linear scalability: The encoding time grows proportionally with the number of records. This is a critical property, as it ensures that SAUL remains a viable encoding scheme even for large-scale record linkage tasks.

While SAUL is arguably slower than both standard Bloom filters and BAD, the performance overhead is a direct consequence of the additional computational steps required to achieve its strong security guarantees. However, we would argue that the performance penalty is easily justified by the increased security, as SAUL is only around 10% slower than BAD. For instance, encoding 100 million records on our hardware and using our implementation of SAUL would take around 3:52 h, which is only 22 minutes longer than using BAD.

6 Conclusion

We identified weaknesses in the Bloom Filter with added diffusion (BAD) encoding scheme, which was previously found to be the only encoding scheme resistant against graph matching attacks (GMAs). In particular, we demonstrated that BAD leaks structural relationships of the plaintext records into the encoded data. Exploiting this weakness, our newly proposed homomorphism-based graph matching attack (H-GMA) achieves re-identification rates of up to 91.6% of records in our experiments, breaking the last remaining secure non-interactive PPRL scheme.

As a remedy, we introduce the Scheme for Anonymous Utility-preserving Linkage (SAUL), a novel encoding scheme specifically designed to prevent homomorphism-based attacks by employing a two-stage process of majority aggregation and XORing of intermediate representations. Theoretical analysis and empirical evaluation confirm that SAUL is not only secure against H-GMA and other known attacks, but also outperforms BAD in terms of linkage quality, making it a secure and practical drop-in replacement for BAD.

Acknowledgments

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

References

[1] Frederik Armknecht, Youzhe Heng, and Rainer Schnell. 2023. Strengthening Privacy-Preserving Record Linkage using Diffusion. *Proceedings on Privacy Enhancing Technologies* 2023 (04 2023), 298–311. https://doi.org/10.56553/popets-2023-0054

[2] An Braeken and Bart Preneel. 2005. On the Algebraic Immunity of Symmetric Boolean Functions. In *Progress in Cryptology - INDOCRYPT 2005*, Subhamoy Maitra, C. E. Veni Madhavan, and Ramarathnam Venkatesan (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 35–48.

[3] Peter Christen, Thilina Ranbaduge, and Rainer Schnell. 2020. *Linking Sensitive Data: Methods and Techniques for Practical Privacy-Preserving Information Sharing.* Springer, Cham. https://doi.org/10.1007/978-3-030-59706-1

[4] Patricia Correll, Anne-Marie Feyer, Thao Phan, Barry Drake, Walid Jammal, Katie Irvine, Adrian Power, Sharon Muir, Shahana Ferdousi, Samantha Moubarak, Yalchin Oytam, James Linden, and Louise Fisher. 2021. Lumos: a statewide linkage programme in Australia integrating general practice data to guide system redesign. *Integrated Healthcare Journal* 3 (05 2021), 1–9. https://doi.org/10.1136/ihj-2021-000074

[5] Ivan P. Fellegi and Alan B. Sunter. 1969. A Theory of Record Linkage. *J. Amer. Statist. Assoc.* 64, 328 (1969), 1183–1210. http://courses.cs.washington.edu/courses/cse590q/04au/papers/Felligi69.pdf

[6] Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable Feature Learning for Networks. arXiv:1607.00653 [cs.SI]

[7] Abel N Kho, John P Cashy, Kathryn L Jackson, Adam R Pah, Satyender Goel, Jörn Boehnke, John Eric Humphries, Scott Duke Kominers, Bala N Hota, Shannon A Sims, Bradley A Malin, Dustin D French, Theresa L Walunas, David O Meltzer, Erin O Kaleba, Roderick C Jones, and William L Galanter. 2015. Design and implementation of a privacy preserving electronic health record linkage tool in Chicago. *Journal of the American Medical Informatics Association* 22, 5 (Jun 2015), 1072–1080. https://doi.org/10.1093/jamia/ocv038

[8] Mehmet Kuzu, Murat Kantarcioglu, Elizabeth Durham, and Bradley Malin. 2011. A Constraint Satisfaction Cryptanalysis of Bloom Filters in Private Record Linkage. In *The 11th Privacy Enhancing Technologies Symposium*, S. Fischer-Hübner and N. Hopper (Eds.). Springer, Berlin, 226–245. https://doi.org/10.1007/978-3-642-22263-4_13

[9] Mitsuru Matsui. 1994. Linear Cryptanalysis Method for DES Cipher. In *Advances in Cryptology — EUROCRYPT '93*, Tor Helleseth (Ed.). Springer Berlin Heidelberg, Berlin, Heidelberg, 386–397.

[10] Jens Meier, Tobias Jakscha, Rainer Schnell, and Günther Heller. 2017. *Technische Dokumentation zur Umsetzung der Pseudonymisierung der PID-Daten für die Module Geburtshilfe und Neonatologie des QS-Verfahrens Perinatalmedizin in der Vertrauensstelle.* IQTIG – Institut für Qualitätssicherung und Transparenz im Gesundheitswesen, Berlin. https://iqtig.org/downloads/spezifikation/2018/v01/TechDok_Verknuepfung_Peri_Neo_V03.pdf

[11] Frank Niedermeyer, Simone Steinmetzer, Martin Kroll, and Rainer Schnell. 2014. Cryptanalysis of Basic Bloom Filters Used for Privacy Preserving Record Linkage. *Journal of Privacy and Confidentiality* 6 (12 2014). https://doi.org/10.29012/jpc.v6i2.640

[12] Robespierre Pita, Clícia Pinto, Samila Sena, Rosemeire Fiaccone, Leila Amorim, Sandra Reis, Mauricio L. Barreto, Spiros Denaxas, and Marcos Ennes Barreto. 2018. On the Accuracy and Scalability of Probabilistic Data Linkage Over the Brazilian 114 Million Cohort. *IEEE Journal of Biomedical and Health Informatics* 22, 2 (2018), 346–353. https://doi.org/10.1109/JBHI.2018.2796941

[13] Thilina Ranbaduge, Peter Christen, and Rainer Schnell. 2020. Secure and Accurate Two-Step Hash Encoding for Privacy-Preserving Record Linkage. In *Advances in Knowledge Discovery and Data Mining*, Hady W. Lauw, Raymond Chi-Wing Wong, Alexandros Ntoulas, Ee-Peng Lim, See-Kiong Ng, and Sinno Jialin Pan (Eds.). Springer International Publishing, Cham, 139–151.

[14] Sean Randall, Adrian P Brown, Anna M Ferrante, and James H Boyd. 2021. Privacy preserving linkage using multiple dynamic match keys. *International Journal of Population Data Science* 4, 1 (Jul. 2021), 1–11. https://doi.org/10.23889/ijpds.v4i1.1094

[15] Kurt Schmidlin, Kerri Clough-Gorr, and Adrian Spoerri. 2015. Privacy Preserving Probabilistic Record Linkage (P3RL): A novel method for linking existing health-related data and maintaining participant confidentiality. *BMC medical research methodology* 15 (05 2015), 46. https://doi.org/10.1186/s12874-015-0038-6

[16] Jochen Schäfer, Frederik Armknecht, and Youzhe Heng. 2024. R+R: Revisiting Graph Matching Attacks on Privacy-Preserving Record Linkage. In *2024 Annual Computer Security Applications Conference (ACSAC)*. IEEE, Honolulu, 699–715. https://doi.org/10.1109/ACSAC63791.2024.00064

[17] D. Smith. 2017. Secure pseudonymisation for privacy-preserving probabilistic record linkage. *Journal of Information Security and Applications* 34 (2017), 271–279. https://doi.org/10.1016/j.jisa.2017.01.002

[18] Alan Stuart and Keith Ord. 2000. *Kendall's advanced theory of statistics. 1 Distribution theory* (6. ed., repr. ed.). Hodder Arnold, London.

[19] Anushka Vidanage, Thilina Ranbaduge, Peter Christen, and Sean Randall. 2020. A Privacy Attack on Multiple Dynamic Match-key based Privacy-Preserving Record Linkage. *International Journal of Population Data Science* 5, 1 (Aug. 2020), 1–13. https://doi.org/10.23889/ijpds.v5i1.1345

[20] Anushka Vidanage, Thilina Ranbaduge, Peter Christen, and Rainer Schnell. 2019. Efficient Pattern Mining Based Cryptanalysis for Privacy-Preserving Record Linkage. In *2019 IEEE 35th International Conference on Data Engineering ICDE 2019*. IEEE, Los Alamitos, 1698–1701. https://doi.org/10.1109/ICDE.2019.00176