


Using Identification Codes in the Two-Party Privacy-Preserving Record Linkage (PPRL)

Yanling Chen

(Volkswagen Infotainment GmbH, Bochum, Germany)

 <https://orcid.org/0000-0003-1603-9121>, yanling.chen@volkswagen-infotainment.com)

Abstract: In this paper, we show the problem of two-party privacy-preserving record linkage (PPRL) can be seen as an identification problem in Information Theory. We propose to apply the identification codes that are designed for identification via channels to the problem of PPRL, due to their advantage in the performance analysis, especially on a quantitative evaluation of the privacy. Note that for the PPRL, linkage quality is typically evaluated experimentally, whilst for privacy, there are so far no commonly accepted privacy measures available that allow an objective evaluation. Our approach of identification code provides an objective evaluation on both linkage quality and privacy based on parameters of employed identification codes.

Keywords: Record Linkage, Identification Code, Privacy

Categories: E.4, H.1.1, H.4.3, G.4

DOI: 10.3897/jucs.167412

1 Introduction

1.1 Privacy-preserving record linkage

Privacy-preserving record linkage (PPRL) addresses the problem of linking records that represent the same individuals across several datasets without revealing sensitive information of the individuals [Christen (2012), Christen et al. (2020)]. So far it has attracted broad research attention. Various linkage protocols have been proposed. Interested readers are referred to a short list of publications on PPRL that includes, but is not limited to [Christen and Verykios (2012), Schnell et al. (2009), Ariel et al. (2015), Vatsalan and Christen (2014), Christen et al. (2020), Nitz and Mandal (2024)].

In general, proposals to PPRL can be classified into those that require a third party for performing the linkage and those that do not. The former are known as ‘three-party protocols’ and the latter as ‘two-party protocols’. In three-party protocols, a (trusted) third party (which we call the ‘linkage unit’) is involved in conducting the linkage, while in two-party protocols only the two database owners participate in the PPRL process. In this paper, we put our focus on the two-party protocols, whilst an overview of the current approaches and challenges for the two-party PPRL has been given in [Chen (2020)].

Generally, two-party protocols start by the two database owners agreeing upon and exchanging any required information, such as parameter settings, preprocessing methods, encoding or encryption methods, and any secret keys that are required, and further proceed by the secure transmission or exchange of encoded or encrypted attribute values to conduct the linkage. The final step is to derive the identified linked records.

The advantage of two-party over three-party protocols is the fact that no database records are shared with any external party, and thus there is no possibility of collusion

between one of the database owners and the linkage unit. However, two-party protocols could require more sophisticated encoding or encryption mechanisms because both database owners know the full details of the agreed parameters or encoding/encryption techniques, and therefore they can potentially perform attacks on the exchanged (encrypted) data between them to infer actual values from each other's data. In other words, the core encryption/encoding techniques need to ensure that each database owner cannot infer any sensitive information (on the non-linked records) from the other database with knowledge of both encrypted data sets and shared system parameters.

More formally, for the privacy analysis, we assume a semi-honest threat model. We say that a two-party PPRL protocol is *secure* in a semi-honest model when neither party can gain any information from the execution of the protocol other than the information gained from the protocol's output (and the other party's input). The semi-honest security model is contrasted to the malicious security model, where the latter allows adversaries to arbitrarily deviate from the specified protocol while attempting to non-consensually gain information from the protocol's execution.

1.2 Identification via Channels

For the standard problem of transmission, the model was introduced by Shannon in his landmark paper [Shannon (1948)] and shown in Fig. 1. The goal is to encode a message in such a way that after it passes through a noisy channel, the message can be successfully decoded at the receiver. It turns out that one can send messages that scale exponentially with the block length and have the error probability of decoding arbitrarily small. For this case, error control coding provides ways of adding redundancy into messages so that the receiver can still determine the sent message correctly despite the noise added during the transmission.

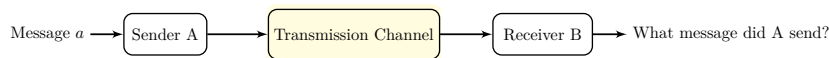


Figure 1: Model for standard transmission problem over channels

For the problem of identification via channels, the model was introduced by Ahlswede and Dueck [Ahlswede and Dueck (1989)] and shown in Fig. 2. Differently from the transmission problem, here the receiver is only interested in testing whether a particular message was sent, but the encoder does not know which message the decoder wants. That is, the encoder sends a message, a , but the decoder would like to know if message b is sent. (Another interpretation in a multi-terminal setting is that, suppose that the sender's intended terminal is a ; upon receiving information broadcast by the sender, each terminal b could identify whether it is the intended recipient.) It turns out that one can design systems such that the number of different messages/terminals one can identify grows doubly exponentially with the block length. For this case, errors are considered in terms of false identification and missed identification, and the idea behind the optimal coding strategy is to map each message into a list of codewords and the encoder selects one randomly; as long as the fraction of the pairwise overlap of these lists is negligibly small, the error probabilities will be negligibly small.

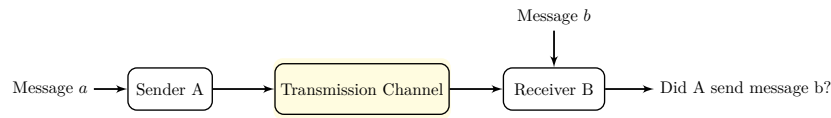


Figure 2: Model for identification problem over channels

1.3 PPRL Seen as an Identification Problem

Immediately we notice the similarity between the problem of identification via channels and the problem of record linkage, especially if we consider the scenario of two-party PPRL with one data holder A as the encoder, the other data holder B as the decoder, and each record corresponding to a message/terminal. See Fig. 3 for the two-party PPRL protocol reformulated as a model of identification via channels. For the record linkage, data holder A sends information about record r_a (e.g., an anonymized form of record r_a) to data holder B. Data holder B tries to identify whether it is a match with record r_b he has (e.g., via comparison between the anonymized form of r_a with a similarly anonymized record r_b). Clearly, once two data holders agree on using an identification code for the two-party PPRL, the encoding procedure is conducted at one data holder to anonymize its records, while the decoding procedure is employed at the other data holder to identify whether the record pair is a match.

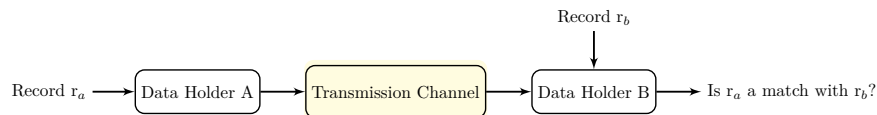


Figure 3: Model for record linkage problem in a 2-party PPRL protocol

1.4 Contribution and organization of the paper

Inspired by the observation of the similar formulation between two problems, in this paper, we propose to use the identification codes that are designed for identification via channels to the problem of two-party privacy-preserving record linkage. Benefiting from the existing results on identification codes in information theory, we show that its applications to the two-party PPRL allow an objective evaluation of both linkage quality and privacy based on parameters of identification codes.

It is worth mentioning that existing constructions of identification codes in information theory mostly aim to be optimal in the sense of identification capacity, which does not immediately indicate an overall optimal performance when using those identification codes in PPRL problems. In fact, the construction of identification codes that are desired for PPRL are those that lead to good performance in linkage quality and privacy and, at the same time, have low implementation complexity and thus scalability.

For the construction of identification codes, in general, we follow the proposal of [Verdú and Wei (1993)] by first constructing an identification plus transmission code (IT code) and then deriving the identification code (ID code) from the IT code. Moreover, the IT code can be constructed by the concatenation of a transmission code and a binary

constant-weight code (BSWC code). Among many existing proposals for the construction of the BSWC code, we consider the proposal in [Kurosawa and Yoshida (1999)], instead of the one proposed in [Verdú and Wei (1993)] (i.e., constructing the BSWC code via concatenation of error-correcting codes, e.g., Reed-Solomon codes, the performance of which is investigated in [Derebeyoğlu et al. (2020), Lengerke et al. (2023), Hefelet al. (2022)]). Experimentally we show that our choice of ID code by using the proposal in [Kurosawa and Yoshida (1999)], i.e., constructing the BSWC code via ϵ -almost strongly universal (ϵ -ASU) hash functions, leads to good linkage performance with acceptable implementation complexity, especially in terms of computational efficiency.

The rest of the paper is organized as follows: in Sec. 2, we introduce the definition of the identification code. In Sec. 3, we discuss the relationship of the different performance measures used in the identification problem and PPRL problem. In Sec. 4, we describe the construction of two concrete ID codes based on two concrete constructions of ϵ -ASU hash functions. In Sec. 5, we present our proposal to apply the identification code to the two-party PPRL. The performance analysis and experimental study are discussed in Sec. 6. Finally, we conclude and point out directions of future research in Sec. 7.

2 Preliminaries

Before proceeding, we provide some definitions that will be useful in the problem setup, especially to distinguish between the definition of codes used for error control and codes for identification. In general, we assume a channel with input alphabet \mathcal{A} and output alphabet \mathcal{B} . We also notice that the scenario of PPRL corresponds to the noiseless channel in which $\mathcal{A} = \mathcal{B}$.

2.1 Transmission code

For the transmission problem, we use (n, L, λ) to denote a transmission code that satisfies

$$\Pr\{s \text{ is decoded at receiver} \mid s \text{ is sent by transmitter}\} \geq 1 - \lambda,$$

for each message s , where each codeword has length n and there are L messages. The rate of an (n, L, λ) transmission code is $\frac{1}{n} \log L$.

2.2 Identification plus Transmission code (IT code)

Han and Verdú [Han and Verdú (1992)] also introduced the model of identification plus transmission code (IT code), where a central station wishes to transmit one of the M messages to one of the N terminals (suppose that codeword $f(a, m)$, of length n , is sent for message m to terminal a). Upon receiving the codeword, each terminal decides whether it is the intended recipient of the message and if so it decodes the message. The decoding reliability of which is measured by (λ_1, λ_2) as follows:

1. for each terminal a :

$$\Pr \left\{ \begin{array}{l} a \text{ decides that it is the intended} \\ \text{recipient, } m \text{ is decoded} \end{array} \middle| \begin{array}{l} a \text{ is the intended recipient,} \\ m \text{ is transmitted} \end{array} \right\} \geq 1 - \lambda_1,$$

2. for any pair of terminals $b \neq a$:

$$\Pr \left\{ b \text{ decides that it is the intended recipient} \mid \begin{array}{l} a \text{ is the intended recipient,} \\ m \text{ is transmitted} \end{array} \right\} \leq \lambda_2,$$

where the probability is taken over all codewords for terminal a in both equations. The *rate-pair* of an $(n, N, M, \lambda_1, \lambda_2)$ IT code is $(\frac{1}{n} \log M, \frac{1}{n} \log \log N)$.

2.3 Identification code (ID code)

Given any $(n, N, M, \lambda_1, \lambda_2)$ IT code, one can immediately construct an $(n, N, \lambda_1, \lambda_2)$ ID code by choosing m randomly over $\{1, \dots, M\}$ in the encoding function $f(a, m)$ for each $a = 1, \dots, N$. We use $(n, N, \lambda_1, \lambda_2)$ to denote the obtained identification code (ID code), which satisfies

1. for each terminal a :

$$\Pr\{a \text{ decides that it is intended} \mid a \text{ is the intended recipient}\} \geq 1 - \lambda_1,$$

2. for any pair of terminals $b \neq a$:

$$\Pr\{b \text{ decides that it is intended} \mid a \text{ is the intended recipient}\} \leq \lambda_2,$$

where the probability is taken over all codewords for terminal a in both equations, each codeword has length n and there are N terminals. The *rate* of an $(n, N, \lambda_1, \lambda_2)$ ID code is $\frac{1}{n} \log \log N$.

3 Relationship between λ_1, λ_2 and precision, recall, specificity

For each encoding and decoding that corresponds to a record comparison in two-party PPRL, the decoding result can be assigned into the following 4 categories: True positives (TP), False positives (FP), True negatives (TN), False negatives (FN). In particular, we have

$$\begin{aligned} \Pr\{\text{TP}\} &= \Pr\{b \text{ decides it is intended} \& b = a \mid a \text{ is the intended recipient}\}; \\ \Pr\{\text{FP}\} &= \Pr\{b \text{ decides it is intended} \& b \neq a \mid a \text{ is the intended recipient}\}; \\ \Pr\{\text{TN}\} &= \Pr\{b \text{ decides it is not intended} \& b \neq a \mid a \text{ is the intended recipient}\}; \\ \Pr\{\text{FN}\} &= \Pr\{b \text{ decides it is not intended} \& b = a \mid a \text{ is the intended recipient}\}. \end{aligned}$$

Note that if an $(n, N, \lambda_1, \lambda_2)$ ID code is employed, we have

$$\begin{aligned} \Pr\{b \text{ decides it is intended} \mid a \text{ is the intended recipient} \& b = a\} &\geq 1 - \lambda_1; \\ \Pr\{b \text{ decides it is intended} \mid a \text{ is the intended recipient} \& b \neq a\} &\leq \lambda_2. \end{aligned}$$

Recall that in the context of record linkage [Christen (2012), Christen et al. (2020)],

$$\begin{aligned} \text{Precision} &= \frac{\Pr\{\text{TP}\}}{\Pr\{\text{TP}\} + \Pr\{\text{FP}\}}; & \text{Recall} &= \frac{\Pr\{\text{TP}\}}{\Pr\{\text{TP}\} + \Pr\{\text{FN}\}}. \\ \text{Specificity} &= \frac{\Pr\{\text{TN}\}}{\Pr\{\text{TN}\} + \Pr\{\text{FP}\}}. \end{aligned}$$

Then for the employed $(n, N, \lambda_1, \lambda_2)$ ID code, it is easy to derive the following relationships (see [Chen (2024)] for a detailed derivation):

$$\text{Precision} \geq \frac{\Pr\{\text{TP}\}}{\Pr\{\text{TP}\} + \lambda_2}; \quad \text{Recall} \geq 1 - \lambda_1; \quad \text{Specificity} \geq 1 - \lambda_2.$$

In general, smaller λ_1, λ_2 imply better precision, recall and specificity scores.

4 Construction of ID codes

4.1 Binary constant-weight code (BCWC)

Definition 1 An (L, N, M, K) binary constant-weight code is a set of N binary strings of length L and Hamming weight M such that the pairwise overlap (maximum number of coincident 1's between any two codewords) does not exceed K .

Any (L, N, M, K) binary constant-weight code can be described by an $N \times M$ incidence matrix on $\{1, \dots, L\}$ s.t. the row $(s(a, 1), \dots, s(a, M))$ gives the locations of the M 1's in the a th codeword, for every $a \in \{1, \dots, N\}$. Define

$$\beta = \frac{\log M}{\log L}, \quad \rho = \frac{\log \log N}{\log L}, \quad \mu = \frac{K}{M},$$

where β is called the *weight factor*, ρ is called the *second-order rate* (as opposed to the first-order rate $\frac{1}{L} \log N$), and μ is called the *overlap fraction* of the binary constant-weight code.

4.2 Construction of ID codes via BCWC

Verdú and Wei [Verdú and Wei (1993), Proposition 1] showed that an IT code can be obtained by concatenating a transmission code with a binary constant-weight code.

For the transmission code, we consider the special case where the underlying channel is noiseless, that is, $\mathcal{A} = \mathcal{B}$ and no errors would occur during the transmission from the sender to the receiver (thus $\lambda = 0$). For such a case, a one-to-one mapping can be used for the encoding and decoding purpose. For instance, taking $\mathcal{A} = \mathcal{B} = \{0, 1\}$ and $n = \lfloor \log_2 L \rfloor$, one can define for this transmission code an encoding function $\phi(\cdot)$ that maps s to the binary representation of $s - 1$ for $s \in \{1, \dots, L\}$ in n bits. Since the channel is noiseless, the decoding function could be simply the inverse mapping $\phi^{-1}(\cdot)$.

According to [Verdú and Wei (1993), Proposition 1], an $(n, N, M, 0, \mu)$ IT code can be obtained by concatenating an $(n, L, 0)$ transmission code with an $(L, N, M, \mu M)$ binary constant-weight code. Furthermore, an $(n, N, 0, \mu)$ ID code could be obtained from the $(n, N, M, 0, \mu)$ IT code, the encoding and decoding of which are described in Algorithm 1 and 2, respectively.

Algorithm 1 Encoding of an $(n, N, 0, \mu)$ ID code from the $(n, N, M, 0, \mu)$ IT code via an $(n, L, 0)$ transmission code with an encoding function $\phi(\cdot)$ & an $(L, N, M, \mu M)$ binary constant-weight code with an incidence matrix S .

Input: Intended terminal a . $\triangleright a \in \{1, \dots, N\}$.
 1: Randomly choose m over $\{1, \dots, M\}$.
 2: Compute codeword $c := \phi(S(a, m))$. $\triangleright \phi(\cdot)$ is the encoding function of the $(n, L, 0)$ transmission code.
 $\triangleright S(a, m)$ is the element of S in the a -th row, m -th column.

Output: Codeword c .

Algorithm 2 Decoding of an $(n, N, 0, \mu)$ ID code from the $(n, N, M, 0, \mu)$ IT code via an $(n, L, 0)$ transmission code with a decoding function $\phi^{-1}(\cdot)$ & an $(L, N, M, \mu M)$ binary constant-weight code with an incidence matrix S .

Input: Terminal b and codeword c' . $\triangleright b \in \{1, \dots, N\}$.
 1: Compute $s' := \phi^{-1}(c')$. $\triangleright \phi^{-1}(\cdot)$ is the decoding function of the $(n, L, 0)$ transmission code.
 2: **if** $s' \in S(b, \star)$ **then** $\triangleright S(b, \star)$ is the set of elements in the b -th row of S .
 3: flag \leftarrow TRUE; $\triangleright b$ declares that it is the intended recipient.
 4: **else**
 5: flag \leftarrow FALSE; $\triangleright b$ declares that it is not the intended recipient.
 6: **end if**

Output: flag.

4.3 Construction of BCWC via ϵ -Almost Strongly Universal Hash Functions

While some explicit constructions of BCWCs (e.g., via concatenation of error correcting codes) were presented in [Verdú and Wei (1993)] that lead to ID codes that are optimal for identification; in [Kurosawa and Yoshida (1999)], different constructions of BCWCs (e.g.: via ϵ -almost strongly universal hash functions) were proposed. In this section, we consider the constructions of BCWCs (and eventually ID codes) proposed in [Kurosawa and Yoshida (1999)].

Let \mathcal{X} and \mathcal{Y} be finite sets such that $|\mathcal{X}| \geq |\mathcal{Y}|$, and \mathcal{H} be a set of functions such that $h : \mathcal{X} \rightarrow \mathcal{Y}$ for each $h \in \mathcal{H}$.

Definition 2 We say that \mathcal{H} is an ϵ -almost strongly universal (ϵ -ASU) class of hash functions provided that the following two conditions are satisfied:

- for any $x \in \mathcal{X}, y \in \mathcal{Y}$, there exist exactly $\frac{|\mathcal{H}|}{|\mathcal{Y}|}$ functions $h \in \mathcal{H}$ such that $h(x) = y$;
- from any two distinct elements $x_1, x_2 \in \mathcal{X}$ and for any two (not necessarily distinct) elements $y_1, y_2 \in \mathcal{Y}$, there exist at most $\epsilon \frac{|\mathcal{H}|}{|\mathcal{Y}|}$ functions $h \in \mathcal{H}$ such that $h(x_i) = y_i, i = 1, 2$.

Let \mathcal{H} be an ϵ -ASU class of hash functions from \mathcal{X} to \mathcal{Y} . The incidence matrix of \mathcal{H}

is an $|\mathcal{X}||\mathcal{Y}| \times |\mathcal{H}|$ binary matrix defined by

$$((x, y), h)\text{-th element} = \mathbb{1}_{\{h(x)=y\}} = \begin{cases} 1, & \text{if } h(x) = y; \\ 0, & \text{otherwise.} \end{cases}$$

Then the incidence matrix of \mathcal{H} is an (L, N, M, K) binary constant-weight code with

$$L = |\mathcal{H}|, \quad N = |\mathcal{X}||\mathcal{Y}|, \quad M = \frac{|\mathcal{H}|}{|\mathcal{Y}|}, \quad K = \epsilon \frac{|\mathcal{H}|}{|\mathcal{Y}|},$$

and the overlap factor $u = \frac{K}{M} = \epsilon$.

4.4 Construction of the ϵ -ASU class of hash functions

Let q be a prime power and let $1 \leq k \leq q$. Let $\mathcal{X} = \{(a_1, \dots, a_k) | a_i \in \mathbb{GF}(q)\}$ and $\mathcal{Y} = \{\text{the elements of } \mathbb{GF}(q)\}$.

4.4.1 ID code \mathcal{C}_1

den Boer [den Boer (1993)] described the following ϵ -ASU class of hash functions from \mathcal{X} to \mathcal{Y} :

Definition 3 For $\forall e_0, e_1 \in \mathbb{GF}(q)$, let

$$h_{(e_0, e_1)}(a_1, \dots, a_k) = e_0 + a_1 e_1 + \dots + a_k e_1^k.$$

Then $\mathcal{G}(q, k) = \{h_{(e_0, e_1)}\}$ is an ϵ -ASU class of hash functions from \mathcal{X} to \mathcal{Y} such that $|\mathcal{G}(q, k)| = q^2$ and $\epsilon = \frac{k}{q}$.

Such an ϵ -ASU class of hash functions implies an $(L, N, M, \mu M)$ binary constant-weight code with

$$L = |\mathcal{G}(q, k)| = q^2, \quad N = |\mathcal{X}||\mathcal{Y}| = q^{k+1}, \quad M = \frac{|\mathcal{G}(q, k)|}{|\mathcal{Y}|} = q, \quad u = \epsilon = \frac{k}{q}.$$

If we consider an $(n, L, 0)$ transmission code with $n = 2$ and $L = q^2$ (i.e., each codeword is of length 2 over $\mathbb{GF}(q)$); and an $(L, N, M, \mu M)$ binary constant-weight code constructed by $\mathcal{G}(q, k)$ as defined above with $L = q^2, N = q^{k+1}, M = q$ and $\mu = \frac{k}{q}$, then we obtain a $(2, q^{k+1}, 0, \frac{k}{q})$ ID code \mathcal{C}_1 according to [Verdú and Wei (1993), Proposition 1]. Note that this code could identify $N = q^{k+1}$ terminals. Each terminal can be indexed by (\mathbf{a}, α) , where $\mathbf{a} = (a_1, \dots, a_k)$ with $a_i \in \mathbb{GF}(q)$ for $i = 1, \dots, k$ and α is an element of $\mathbb{GF}(q)$.

The encoding and decoding of this specific ID code instance are described in Algorithm 3 and 4, respectively. Compared with the general ID codes constructed by using binary constant-weight codes, this specific construction offers a few advantages. It provides efficient encoding and decoding algorithms (i.e., reduced computational complexity for each encoding-decoding procedure). Besides, there is also no need to store the N by M incidence matrix S at both the encoder and decoder (thus saving the storage cost at both sides).

Algorithm 3 Encoding of the $\mathcal{C}_1 : (2, q^{k+1}, 0, \frac{k}{q})$ ID code

Input: Intended terminal (\mathbf{a}, α) . $\triangleright \mathbf{a} = (a_1, \dots, a_k) \in \mathbb{GF}(q)^k$ and $\alpha \in \mathbb{GF}(q)$.
 1: Randomly choose e_1 over $\mathbb{GF}(q)$. \triangleright A random choice from those q choices of (e_0, e_1) s.t. $h_{(e_0, e_1)}(\mathbf{a}) = \alpha$.
 2: Compute e_0 s.t. $e_0 + a_1 e_1 + \dots + a_k e_1^k = \alpha$.
Output: Codeword (e_0, e_1) . $\triangleright e_0, e_1 \in \mathbb{GF}(q)$.

Algorithm 4 Decoding of the $\mathcal{C}_1 : (2, q^{k+1}, 0, \frac{k}{q})$ ID code

Input: Terminal (\mathbf{b}, β) and codeword (e_0, e_1) . $\triangleright \mathbf{b} = (b_1, \dots, b_k) \in \mathbb{GF}(q)^k$ and $\beta \in \mathbb{GF}(q)$.
 1: **if** $e_0 + b_1 e_1 + \dots + b_k e_1^k = \beta$ **then** \triangleright Check whether $h_{(e_0, e_1)}(\mathbf{b}) = \beta$ hold or not.
 2: flag \leftarrow TRUE; $\triangleright (\mathbf{b}, \beta)$ declares that it is the intended recipient.
 3: **else**
 4: flag \leftarrow FALSE; $\triangleright (\mathbf{b}, \beta)$ declares that it is not the intended recipient.
 5: **end if**
Output: flag.

4.4.2 ID code \mathcal{C}_2

Stinson [Stinson (1994)] showed a composition construction of an ϵ -ASU class of hash functions as follows.

Proposition 1 [Stinson (1994), Theorem 5.5] *Let $C = (n, |\mathcal{C}|, d)$ be an error-correcting code over an alphabet \mathcal{X} . Let \mathcal{H} be an ϵ -ASU class of hash functions from \mathcal{X} to \mathcal{Y} . Then for all i with $1 \leq i \leq n$ and $\forall h \in \mathcal{H}$, define a hash function $g_{(i, h)} : \mathcal{C} \rightarrow \mathcal{Y}$ by the rule*

$$g_{(i, h)}(x) = h(\text{the } i\text{-th symbol of the } x\text{-th codeword of } C)$$

Let $\mathcal{H}_C = \{g_{(i, h)}\}$. \mathcal{H}_C is an $\tilde{\epsilon}$ -ASU class of hash functions from \mathcal{C} and \mathcal{Y} such that

$$\tilde{\epsilon} = \epsilon + 1 - \frac{d}{n}; \quad |\mathcal{H}_C| = n|\mathcal{H}|.$$

Take C by a $[q^k, q^t]$ Reed-Solomon code over $\mathbb{GF}(q^k)$ s.t.

- the length of a codeword is $n = q^k$;
- the number of codewords is $|\mathcal{C}| = (q^k)^{q^t}$;
- the minimum Hamming distance is $d = q^k - q^t + 1$.

Let $\mathbf{G} = (G_1, \dots, G_{q^k})$ be a generator matrix $\mathbf{G} = (G_1, \dots, G_{q^k})$ of C .

Let \mathcal{H} be the $\frac{k}{q}$ -ASU class of hash functions as defined in Definition 3.

Then, according to Proposition 1, we obtain a $\tilde{\epsilon}$ -ASU class of hash functions from \mathcal{C} to \mathcal{Y} such that

$$\tilde{\epsilon} = \frac{k}{q} + 1 - \frac{d}{n} = \frac{k}{q} + \frac{1}{q^{k-t}} - \frac{1}{q^k}; \quad |\mathcal{H}_C| = nq^2 = q^{k+2}.$$

Furthermore, such an $\tilde{\epsilon}$ -ASU class of hash functions implies an (L, N, M, K) binary constant-weight code with

- $L = |\mathcal{H}_C| = q^{k+2}$,
- $N = |\mathcal{C}||\mathcal{Y}| = q^{kq^t+1}$,
- $M = \frac{|\mathcal{H}_C|}{|\mathcal{Y}|} = q^{k+1}$,
- $K = \tilde{\epsilon} \frac{|\mathcal{H}_C|}{|\mathcal{Y}|} = kq^k + q^{t+1} - q$,

where $t < k < q$ and q is a prime. It is easy to calculate this binary constant-weight code has the overlap fraction μ as follows:

$$\mu = \frac{K}{M} = \tilde{\epsilon} = \frac{k}{q} + \frac{1}{q^{k-t}} - \frac{1}{q^k}.$$

If we consider an $(n, L, 0)$ transmission code with $n = k+2$ and $L = q^{k+2}$ (i.e., each codeword is of length $k+2$ over $\mathbb{GF}(q)$); and an $(L, N, M, \mu M)$ binary constant-weight code as defined above with $L = q^{k+2}$, $N = q^{kq^t+1}$, $M = q^{k+1}$ and $\mu = \frac{k}{q} + \frac{1}{q^{k-t}} - \frac{1}{q^k}$, then we obtain an $(k+2, q^{kq^t+1}, 0, \frac{k}{q} + \frac{1}{q^{k-t}} - \frac{1}{q^k})$ ID code \mathcal{C}_2 according to [Verdú and Wei (1993), Proposition 1]. Note that this code could identify $N = q^{kq^t+1}$ terminals. Each terminal can be indexed by (\mathbf{a}, α) , where $\mathbf{a} = (a_1, \dots, a_{q^t})$ with $a_i \in \mathbb{GF}(q^k)$ for $i = 1, \dots, q^t$ and α is an element of $\mathbb{GF}(q)$. The encoding and decoding of this specific ID code instance are described in Algorithm 5 and 6, respectively.

Algorithm 5 Encoding of the $\mathcal{C}_2 : (k+2, q^{kq^t+1}, 0, \frac{k}{q} + \frac{1}{q^{k-t}} - \frac{1}{q^k})$ ID code

Input: Intended terminal (\mathbf{a}, α) , generator matrix $\mathbf{G} = (G_1, \dots, G_{q^k})$ of $[q^k, q^t]$ Reed-Solomon code C .
 $\triangleright \mathbf{a} = (a_1, \dots, a_{q^t}) \in \mathbb{GF}(q^k)^{q^t}$ and $\alpha \in \mathbb{GF}(q)$.

- 1: Randomly choose $i \in \mathbb{GF}(q^k)$.
- 2: Calculate the i -th symbol of \mathbf{a} 's codeword in C , by $x = \mathbf{a}G_i$ over $\mathbb{GF}(q^k)$, and denote its vector representation over $\mathbb{GF}(q)$ as $x = (x_1, \dots, x_k)$.
- 3: Randomly choose e_1 over $\mathbb{GF}(q)$.
- 4: Compute e_0 s.t. $e_0 + x_1e_1 + \dots + x_ke_1^k = \alpha$.

Output: Codeword (i, e_0, e_1) . $\triangleright i \in \mathbb{GF}(q^k)$ and $e_0, e_1 \in \mathbb{GF}(q)$.

Algorithm 6 Decoding of the $\mathcal{C}_2 : (k + 2, q^{kq^t+1}, 0, \frac{k}{q} + \frac{1}{q^{k-t}} - \frac{1}{q^k})$ ID code

Input: Terminal (\mathbf{b}, β) , codeword (i, e_0, e_1) , and generator matrix $\mathbf{G} = (G_1, \dots, G_{q^k})$ of $[q^k, q^t]$ Reed-Solomon code C . $\triangleright \mathbf{b} = (b_1, \dots, b_{q^t}) \in \mathbb{GF}(q^k)^{q^t}$ and $\beta \in \mathbb{GF}(q)$.

- 1: Calculate the i -th symbol of \mathbf{b} 's codeword in C , by $y = \mathbf{b}G_i$ over $\mathbb{GF}(q^k)$, and denote its vector representation over $\mathbb{GF}(q)$ as $y = (y_1, \dots, y_k)$.
- 2: **if** $e_0 + y_1e_1 + \dots + y_ke_1^k == \beta$ **then**
- 3: flag \leftarrow TRUE; $\triangleright (\mathbf{b}, \beta)$ declares that it is the intended recipient.
- 4: **else**
- 5: flag \leftarrow FALSE; $\triangleright (\mathbf{b}, \beta)$ declares that it is not the intended recipient.
- 6: **end if**

Output: flag.

5 Using ID codes in two-party PPRL

For a two-party PPRL, suppose that data holder A sends the anonymized data set to data holder B; and data holder B conducts the linkage based on the anonymized data set from A and its own data set. Then an ID code could be used in a two-party PPRL protocol by considering each record as a terminal. In more detail, the record anonymization procedure at the data holder A for each record is corresponding to the encoding procedure by taking the record as input, whilst the record linkage procedure at the data holder B for each record pair is corresponding to the decoding procedure by taking the received codeword and record at B as input. The decoding outputs a TRUE or FALSE to flag a match or non-match of the record pair.

More specifically, an $(n, N, 0, \mu)$ ID code can be used to link any two records from N different entities. And, for each record, a codeword of length n needs to be transmitted from data holder A to data holder B. According to the discussion in Sec. 3, a simple application of the code (using a single match key as shown in [Chen (2024)]) offers a performance for the two-party PPRL with Recall = 1 and a Specificity $\geq 1 - \mu$.

5.1 Using multiple match keys

Note that inconsistency between matching variables can occur in a number of different forms. A single match key alone might not be able to resolve all the inconsistencies that occur between records belonging to the same individual or identify every individual uniquely. Here we can use multiple match keys, each made up of different combinations of personal identifiers and designed to resolve a certain type of inconsistency that often occurs between records belonging to the same individual. Furthermore, we can use an OR-construction of the linkage scheme that is based on a unique match key, as the one proposed by the Office for National Statistics UK (ONS) [ONS (2013)]. Clearly, for the sake of the linkage performance, match keys should be carefully chosen so that they retain a high level of uniqueness for the majority of records to be matched and at the same time eliminate some of the discrepancies between matching pairs. Interested readers are referred to [ONS (2013), Figure 5] for the structure of the possible match keys and the uniqueness of those keys.

Assume that both database owners A and B have agreed upon a series of multiple match keys (assume in total p different match key constructions), and an ID code for the anonymization and linkage in the two-party protocol. (Note that multiple ID codes could be used, for instance, one ID code for one specific match key construction.) For simplicity, we only detail in the following the scheme with one agreed ID codes. Besides, we assume that the data sets at both data holders have been deduplicated.

Furthermore, we assume an $(n, N, 0, \mu)$ ID code is known to both data holder A and data holder B. And, this ID code is obtained by concatenating an $(n, L, 0)$ transmission code with an $(L, N, M, \mu M)$ binary constant-weight code (that is described by the N by M incidence matrix S).

The two-party protocol using multiple match keys based on one ID code can be described as follows.

At data holder A,

- for each record i , the quasi-identifiers are transformed into a series of match keys, i.e., $mk_{i,1}, \dots, mk_{i,p}$. (Note that some of the match keys might be impossible due to the missing data problem.)

Then for each match key construction j , where $j = 1, \dots, p$, remove the empty or duplicated match keys and only keep the unique ones.

Now suppose that $\{i_1, \dots, i_l\}$ is a list of indexes, indicating under which constructions the multiple match keys are generated and unique for record i . (Note that l could be different for each record at data holder A.)

To apply the ID $(n, N, 0, \mu)$ code, $mk_{i,j} \in \{1, \dots, N\}$ for $j = i_1, \dots, i_l$ are prepared.

- Consider record i . For each $j \in \{i_1, \dots, i_l\}$, taking $mk_{i,j}$ as the input to the encoding procedure of the agreed $(n, N, 0, \mu)$ ID code (see Algorithm 1), each output gives an anonymized form of the record i . More specifically, $S(mk_{i,j}, r_{i,j})$ is the output if taking the encoding and decoding function of the $(n, L, 0)$ transmission code to be the identity function (here $r_{i,j}$ is randomly chosen over $\{1, \dots, M\}$ in the encoding procedure).

For each record i , in total l anonymized forms, i.e., $\{(j, S(mk_{i,j}, r_{i,j})) | j \in \{i_1, \dots, i_l\}\}$, are generated using l different match keys.

After this is done, data holder A sends the list of $\{(j, S(r_{i,j}, mk_{i,j})) | j \in \{i_1, \dots, i_l\}\}$ to data holder B. Since $S(mk_{i,j}, r_{i,j})$ takes values over $\{1, \dots, L\}$ and j takes values over $\{1, \dots, p\}$, for each record, maximal $p \cdot \lfloor \log_2 pL \rfloor$ bits need to be transmitted to data holder B for linkage purpose.

At the data holder B, upon receiving from A the list $\{(j, S(r_{i,j}, mk_{i,j})) | j \in \{i_1, \dots, i_l\}\}$ as an anonymized version of record r_A , B tries to decide whether r_A is a match or not with its record r_B . Due to the OR-construction of this scheme, the matched records could be identified due to possessing at least one private match key. The detailed steps are given as follows:

- Recall that $\{i_1, \dots, i_l\}$ indicates the specific match key constructions used by record r_A .

Accordingly multiple match keys under the same constructions are generated for record r_B .

We suppose that there are in total l' match keys are generated, where $l' \leq l$. (Note that $l' \leq l$ may happen due to the missing data problem in the data set at B).

- For these l' match keys, a match will be identified if there is at least one indexed by j' , under the j' -th construction (from a total of p different constructions of match keys), the match key for record r_B is $mk'_{j'}$, and $mk'_{j'} = mk_{i,j'}$.

In particular, data holder B employs the decoding procedure (as described in Algorithm 2) of the agreed $(n, N, 0, \mu)$ ID code. If $mk'_{j'} = mk_{i,j'}$, then it is clear that $S(mk_{i,j'}, r_{i,j'}) \in S(mk'_{j'}, \star)$, where $S(mk'_{j'}, \star)$ is the set of elements in the $mk'_{j'}$ -th row of S . The decoding algorithm will return $\text{flag} = \text{TRUE}$ and this results in a match.

In general, this multiple match keys scheme based on one ID code has a Recall = 1 and Specificity $\geq (1 - p \cdot \mu)$.

5.2 Some observations

For the proposed scheme, we have the following observations:

- *Record-level flexibility* with the multiple match keys generation: once the total number and the different constructions of the match keys are agreed upon between two data holders, data holder A has the flexibility to generate a certain number of match keys according to its record-level data quality to facilitate the linkage performance.
- *Probabilistic encryption* of records: in the two-party PPRL protocol based on the ID code, M different values of $r_{i,j}$ could be randomly chosen for each specific match key construction.
- *Low communication cost*: data holder A just needs to send the encrypted data set to data holder B once. Differently from the two-party protocols that are based on multi-party secure computation techniques, this is a one-shot game with a low communication cost.
- *Identification of match or non-match*: note that the match keys under p constructions can be pre-computed at each data holder. Using the ID codes, match and non match are indicated by a verification equation.
- *Minimal leakage beyond the matching status*: using the ID code, only match and non-match are derived between records, without involving the calculation of similarity tables or the distance profiles between records (which may leak a significant amount of information and are susceptible to frequency attacks).
- *A primitive for multi-party PPRL*: this scheme could conveniently serve as a fundamental element in a multi-party PPRL scheme without a linkage unit.
- *Further possibilities to improve the performance*: this scheme, using one ID code with multiple match keys, can be extended to use different ID codes for different match keys to reduce the transmission cost and improve the overall specificity.

6 Performance analysis

For simplicity of the performance analysis, in this section, we focus on the scenario of applying ID code to two-party PPRL using a unique match key. Assume data holder A and data holder B each hold a set of records, independently chosen from the population. To conduct the record linkage, data holder A and data holder B agree on using an $(n, N, 0, \mu)$ ID code, which is obtained by concatenating an $(n, L, 0)$ transmission code with an $(L, N, M, \mu M)$ binary constant-weight code (that is described by the N by M incidence matrix S). Straightforwardly, we have the following interpretation of the parameters (if applying this code to 2-party PPRL with a single match key):

- N : the maximal number of entities that can be identified, i.e., *identification capacity*.
- n : the length of the codeword, i.e., the number of digits that data holder A needs to transmit to data holder B for each record, i.e., *transmission cost*.
- M : the maximal number of anonymized forms that can be generated for each record, i.e., *randomness for anonymization*.
- μ : probability of false identification that leads to Precision $\geq \frac{\text{Pr}\{\text{TP}\}}{\text{Pr}\{\text{TP}\} + \mu}$ and Specificity $\geq 1 - \mu$.

Let \mathcal{C} denote the $(n, N, 0, \mu)$ ID code shared at both data holders. For each record r_A , where $r_A \in \{1, \dots, N\}$, data holder A sends a codeword $c(r_A)$, which is an anonymized form of r_A , to data holder B. A natural question is that, how much information does data holder B gain on record r_A by receiving $c(r_A)$? This *information gain* at data holder B is measured by

$$H(r_A|r_B, \mathcal{C}) - H(r_A|c(r_A), r_B, \mathcal{C}),$$

where $H(\cdot)$ is the entropy function. In other words, this is the *information leakage* about r_A from data holder A. The less it is, the better the privacy. Besides, a normalized definition is the *relative information gain* that is measured by

$$\frac{H(r_A|r_B, \mathcal{C}) - H(r_A|c(r_A), r_B, \mathcal{C})}{H(r_A|\mathcal{C})} \quad \text{or} \quad \frac{H(r_A|r_B, \mathcal{C}) - H(r_A|c(r_A), r_B, \mathcal{C})}{\log_2 N}.$$

Proposition 2 *The information gain at data holder B is $\leq \log_2 \frac{L}{M}$ bits.*

First we note that

$$\begin{aligned} H(r_A|c(r_A), r_B, \mathcal{C}) &= H(r_A, c(r_A)|r_B, \mathcal{C}) - H(c(r_A)|r_B, \mathcal{C}) \\ &= H(r_A|r_B, \mathcal{C}) + H(c(r_A)|r_A, r_B, \mathcal{C}) - H(c(r_A)|r_B, \mathcal{C}). \end{aligned}$$

Then the information gain at data holder B is

$$\begin{aligned} H(r_A|r_B, \mathcal{C}) - H(r_A|c(r_A), r_B, \mathcal{C}) &= H(c(r_A)|r_B, \mathcal{C}) - H(c(r_A)|r_A, r_B, \mathcal{C}) \\ &= H(c(r_A)|r_B, \mathcal{C}) - H(c(r_A)|r_A, \mathcal{C}) \\ &\leq \log_2 L - \log_2 M, \end{aligned}$$

where the last inequality is because

1. $H(c(r_A)|r_A, \mathcal{C}) = \log_2 M$. This is because given the code \mathcal{C} (with incidence matrix S), $c(r_A)$ is chosen randomly from the locations of M 1's in the row of matrix S that is corresponding to r_A ;
2. $H(c(r_A)|r_B, \mathcal{C}) \leq H(c(r_A)) \leq \log_2 L$, since conditioning reduces entropy and $c(r_A)$ takes value over $\{1, \dots, L\}$.

So the information gain at data holder B is $\leq \log_2 \frac{L}{M}$ bits and the relative information gain is $\leq \frac{\log_2 L/M}{\log_2 N}$.

6.1 Shared parameters

To apply an ID code to a two-party PPRL protocol, the ID code needs to be shared by data holder A and data holder B for the purpose of the successful linkage. For the ID code is constructed via an $(L, N, M, \mu M)$ binary constant-weight code that is described by an N by M incidence matrix S (as discussed in Sec. 4), the storage cost for the incidence matrix S could be expensive, especially if both data holders have to store the codebook and when N is large (which is supposed to be at least as large as the size of the merged data set A and B).

Some specific constructions of the $(L, N, M, \mu M)$ binary constant-weight codes could avoid such problems. For instance, the construction via the ϵ -ASU class of universal hash functions as discussed in Sec. 4.4.1 is an attractive option, with which there is no need to store the incidence matrix S at both data holders to facilitate the encoding and decoding procedures (see Algorithms 3 and 4; S is not needed for both the encoding and decoding procedures). However, we notice that the construction involving a Reed-Solomon code C , as discussed in Sec. 4.4.2 requires that both data holders share the generator matrix \mathbf{G} of C in $\mathbb{GF}(q^k)$ (see Algorithms 5 and 6), which can be costly in memory to produce and easily become too large to handle, as being noticed in [Derebeyoğlu et al. (2020)].

6.2 Computational cost

For the ID code constructed via an $(L, N, M, \mu M)$ binary constant-weight code that is described by an N by M incidence matrix S (as discussed in Sec. 4), the computation cost for the anonymization or linkage could be expensive, especially when M is large.

Again, some specific constructions of the $(L, N, M, \mu M)$ binary constant-weight codes could offer efficient encoding and decoding of the ID code (and thus efficient anonymization and linkage). Here the construction via the ϵ -ASU class of universal hash functions as discussed in Sec. 4.4.1 is again an attractive option also in this aspect. Especially for the decoding procedure, instead of checking whether the received codeword belongs to a set of M elements, only one equality needs to be checked (see Algorithms 2 and 4). However, we notice that the construction involving a Reed-Solomon code C as discussed in Sec. 4.4.2 requires the calculation in extension field $\mathbb{GF}(q^k)$ (see Algorithms 5 and 6), which can be very computationally expensive, as being noticed in [Derebeyoğlu et al. (2020)].

6.3 Some concrete choices of ID codes

Consider the following two concrete ID code families as we discussed in Sec. 4.4:

1. \mathcal{C}_1 : the $(2, q^{k+1}, 0, \frac{k}{q})$ ID code;
2. \mathcal{C}_2 : the $(k+2, q^{kq^t+1}, 0, \mu)$ ID code with $\mu = \frac{k}{q} + \frac{1}{q^{k-t}} - \frac{1}{q^k}$ and $t < k$.

Some observations on performance evaluation can be made as shown in Table 1.

Performance	$(2, q^{k+1}, 0, \frac{k}{q})$ ID code	$(k+2, q^{kq^t+1}, 0, \frac{k}{q} + \frac{1}{q^{k-t}} - \frac{1}{q^k})$ ID code
Identification capacity	q^{k+1}	q^{kq^t+1}
Transmission cost	$2 \log_2 q$ bits	$(k+2) \log_2 q$ bits
Randomness for anonymization	$\log_2 q$ bits	$(k+1) \log_2 q$ bits
Prob. missed identification	0	0
Prob. false identification	$\leq \frac{k}{q}$	$\leq \frac{k}{q} + \frac{1}{q^{k-t}} - \frac{1}{q^k}$
Information gain	$\leq \log_2 q$ bits	$\leq \log_2 q$ bits
Relative information gain	$\leq \frac{1}{k+1}$	$\leq \frac{1}{kq^t+1}$

Table 1: Performance of \mathcal{C}_1 and \mathcal{C}_2 in 2-party PPRL with unique match key

If the records are uniformly distributed over $\{1, \dots, N\}$, then code \mathcal{C}_1 offers a relative information gain $\leq \frac{1}{k+1}$; whilst code \mathcal{C}_2 offers a relative information gain $\leq \frac{1}{kq^t+1}$.

It is worth mentioning that the probability of false identification μ is the probability to link a random pair of records erroneously. Suppose that the data holder A has a dataset of n_A records, while data holder B has a dataset of n_B records, and there are n_M true matches between these two datasets. Then an estimate for the number of erroneously linked pairs of records is given by $u \cdot (n_A - n_M) \cdot (n_B - n_M)$ and upper bounded by $u \cdot n_A n_B$. For instance, if ID code \mathcal{C}_1 is used for the record linkage, to obtain a low homonym error rate, (k, q) can be chosen such that $\frac{k}{q} \cdot n_A n_B \leq 1$, which leads to $q \geq k n_A n_B$.

6.4 Comparison to hash-based two-party PPRL

6.4.1 Data

We are using the ‘voter ID data’ that is available at <https://dl.ncsbe.gov/index.html?prefix=data/>. The data sets at data holder A and B are generated by independently sampling the ‘voter ID data’ with sampling size ranging from 500 to 2500 records.

6.4.2 Linkage

For comparison purposes, the first and last name attributes are merged into a single string per record. The following 3 methods are considered:

- *plain text*, i.e., data holder A sends the strings in plain text to data holder B;
- *Hash*, i.e., data holder A hashes the strings and then sends them to data holder B;
- *ID code*, i.e., data holder A applies the ID code to (the hash of) each string and then sends the coded version to data holder B.

The experiment is set up to simply compare the anonymization time by using hash and ID code and the linkage time by using these three different methods.

6.4.3 Implementation details

All 3 different encoding methods and linkages are handled using R 4.0.1. In particular, the R package `fastdigest` [Becker and Jenkin (2015)] is used to create 128-bit hashes of randomly drawn strings. For the ID code, we choose different (k, q) values in code \mathcal{C}_1 to illustrate how they impact on the performance.

6.4.4 Linkage quality measures

For simplicity, we take the plain text comparison as the golden standard and only consider the exact matching (with a single match key).

Setting different parameters (k, q) , one can obtain different ID codes. As one can see from Fig. 4, q plays an important role in the linkage quality, which is reflected by MPR (mean of precision and recall). In general, the larger is q , the better the MPR. Especially, the largest choice, $q = 82589933$, is larger than $kn_A n_B$ for the choices of k, n_A, n_B in the experiments. As expected, it gives the best performance on the linkage quality. Besides, we also notice that increasing k does not imply an improve on MPR, although it could potentially improve the privacy (in terms of relative information gain) as given in Table 1.

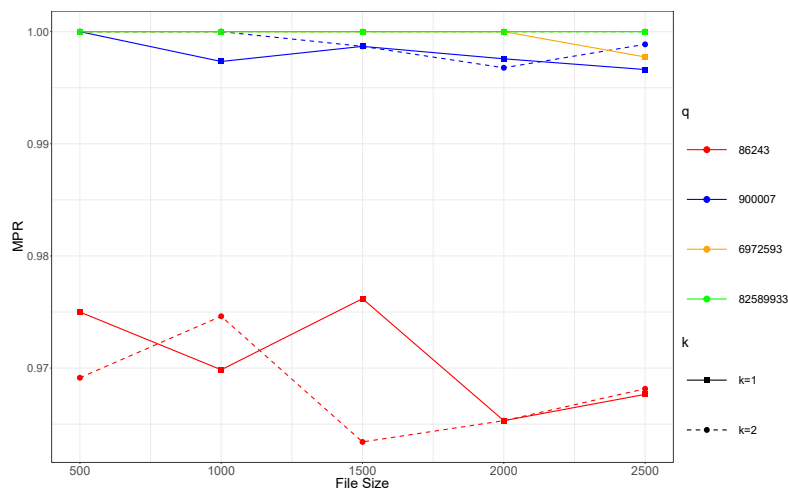


Figure 4: MPR for ID code with different (q, k) settings

6.4.5 Complexity

In general, it takes data holder A (whose task is mainly the anonymization, which has a linear complexity) much less time than data holder B (whose task is mainly the linkage, which has a square complexity) in a two-party PPRL protocol. For each data holder who has a data set with around 1000 records, anonymization takes about $10^{-4} \sim 10^{-3}$ minutes, as one can see from Fig. 5; while linkage takes about $10^{-2} \sim 10^{-1}$ minutes, as

one can see from Fig. 6 and 7. Now let us consider only the hash and ID code. As one can observe from Fig. 5 and Fig. 6, for both the anonymization and the linkage phase, using hashing (especially with pre-computation of the hash values of the records) is more time efficient than using ID codes. Nevertheless, it is worth mentioning that if the pre-computation of the hash values is not employed in the hash-based method, then ID code verification, i.e., linkage, can be faster, as shown in Fig. 7.

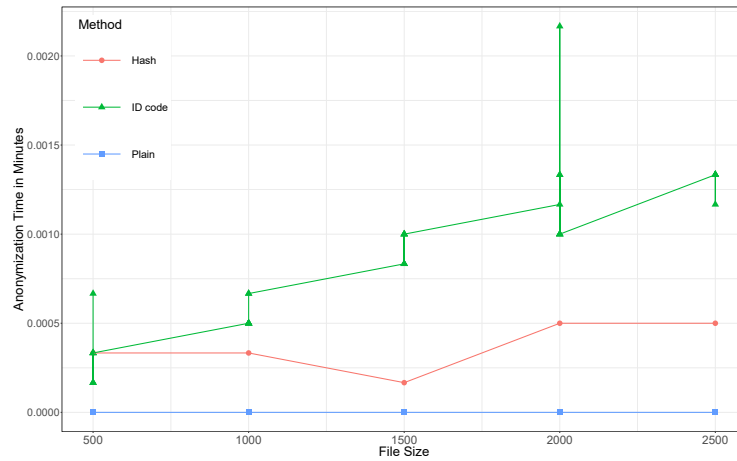


Figure 5: Comparison on Anonymization Time

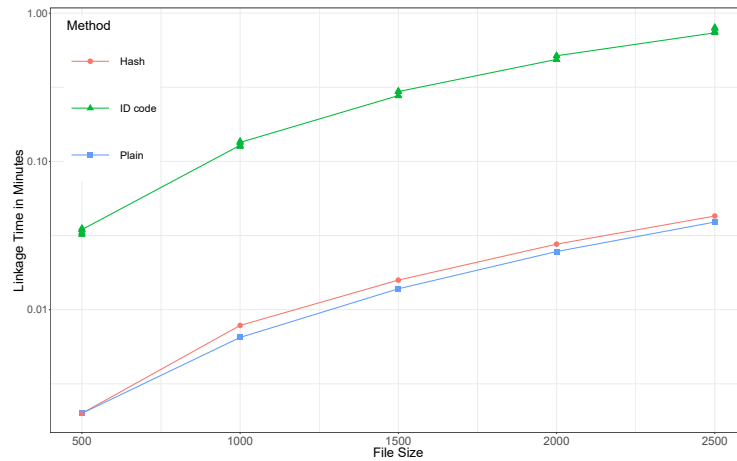


Figure 6: Comparison of Linkage Time (Hash-based Method - with Precomputation)

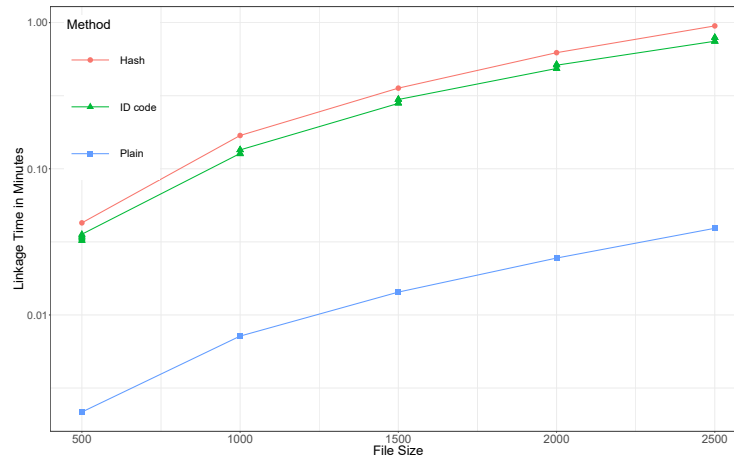


Figure 7: Comparison of Linkage Time (Hash-based Method - Without Precomputation)

6.4.6 Dictionary attack

In this section, we discuss the security of the hash-based approach and the ID code based approach against a dictionary attack.

In a dictionary attack, it is assumed that the attacker

- has access to an encoded database; and
- knows the encoding method and all relevant parameters.

Using publicly available data, ideally a large database covering a full population, the attacker can apply the encoding method on all values in the public database to see if any generated value matches with a value in the encoded database. In this way, the attacker learns the plaintext of the matching record in the encoded database.

It is easy to see that the hash-based encoding method, if without involving any secret key in the encoding process, is not secure against a dictionary attack [Christen et al. (2020)]. While using the ID-code based method as a keyless approach, the same record can be encoded into different values (e.g. q different values if the ID code C_1 is used). Moreover, the ID-code-based approach allows a tiny probability of false identification (e.g., up-bounded by k/q if the ID code C_1 is used). In a dictionary attack, this means that there could be more than one record from the publicly available database that matches to one encoded record in the encoded database, (especially if the number of records in the full population is larger than q/k in case that the ID code C_1 is used), which makes the unique identification of the encoded record difficult or even impossible. Nevertheless, we notice that a dictionary attack can be still effective, especially in narrowing down the possible plaintexts of the matching records.

Nevertheless, it is worth mentioning that there are alternatives to the hash-based method that involve a secret key pre-shared between database owners that could make it a nearly impossible task for an attacker to conduct a dictionary attack. One alternative is simply to concatenate each record with a secret key before the hashing takes place, as mentioned in [Christen et al. (2020)], where the security relies on the preimage resistance of the hash function and the secrecy of the key. Another alternative (suggested by Prof.

Frederik Armknecht), which involves a pre-shared secret key and allows encoding each record differently, is described as follows:

Let mk denote the match key formulation of the record. We assume that data holders A and B share a key key and use some secure block cipher like AES in a secure mode of operation, e.g., CBC.

As described in Algorithm 7, data holder A does the following:

- For each mk , data holder A chooses a random value r as the initial value.
- Compute $c = \text{Enc}(key, r, \text{Hash}(mk))$, i.e., encrypt $\text{Hash}(mk)$ under key key and an initial random value r .
- Send the pair (r, c) to data holder B.

Algorithm 7 Encoding of the Hash-based Alternative with a Pre-shared Key

Input: Record in match key transformation as mk , pre-shared key key .

- 1: Randomly choose a random number r .
- 2: Compute $c = \text{Enc}(key, r, \text{Hash}(mk))$. $\triangleright \text{Enc}(\cdot)$ could be the encryption of AES in a secure mode of operation, e.g., CBC.

Output: Codeword (r, c) .

As described in Algorithm 8, data holder B does the following:

- For each pair (r, c) it receives from A, data holder B computes $h = \text{Dec}(key, r, c)$.
- Checks if h appears in its database.

Note that we assume that data holder B has already prepared a sorted list of $\text{Hash}(mk)$ for all records in its database, as in this way the linkage can be made more time-efficient. That is, for each h it computes, it checks if h can be found in the list. If it is found in the list, then the respective record from data holder A is also contained in the database at data holder B with overwhelming probability. Otherwise, we declare that there is no match in database B to the respective record from data holder A. The security of this alternative scheme relies on the preimage resistance of the hash function and the security of the encryption procedure.

Clearly, involving a secret key pre-shared between database owners serves as an effective countermeasure against a dictionary attack. This idea can also be conveniently used to strengthen the ID code-based approach.

7 Conclusion

In this paper, we show that the identification problem over channels in information theory also models the record linkage problem. This observation inspires us to apply the identification codes to the privacy-preserving record linkage problem. Note for the PPR, linkage quality is typically evaluated experimentally, and for privacy, there are no clearly stated privacy criteria in the legislation [GDPR], and so far no commonly

Algorithm 8 Decoding of the Hash-based Alternative with a Pre-shared Key

Input: Record in match key transformation as mk , pre-shared key key and (r, c) .

- 1: Compute $h = \text{Dec}(key, r, c)$ ▷ $\text{Dec}(\cdot)$ is the decryption function of $\text{Enc}(\cdot)$.
- 2: **if** $h == \text{Hash}(mk)$ **then**
- 3: $\text{flag} \leftarrow \text{TRUE}$; ▷ It is declared as a match.
- 4: **else**
- 5: $\text{flag} \leftarrow \text{FALSE}$; ▷ It is declared as a mismatch.
- 6: **end if**

Output: flag .

accepted privacy measures available that allow an objective evaluation. Our approach of identification code could provide an objective evaluation on both linkage quality (by diminishing the probability of false identification) and privacy (by up bounding the relative information gain) based on parameters of the concrete identification codes implemented.

Furthermore, we show some concrete construction of identification codes and demonstrate the advantage of their application in PPRL over the classical hash-based approaches. Our numeric results suggest that a good linkage performance can be achieved; the computational efficiency is comparable to the hash-based method; and it has an advantage over the hashed-based method against the dictionary attack (while both schemes can be strengthened by introducing a pre-shared key) and in the transmission cost (while the identification code can be considered as a light-weighted hash).

We also notice that, despite the formulation similarity between the PPRL and the identification problem, there are different interests from the respective communities. For the identification problem, it is addressed with more focus on the fundamental limits [Ahlsweede and Dueck (1989), Ahlsweede and Zhang (1995)] (e.g., on capacity-achieving construction of identification codes), while the implementation complexity often is the bottleneck that hinders the application of such codes in practical systems [Derebeyoğlu et al. (2020), Lengerke et al. (2023), Hefele et al. (2022)]. Although the capacity-achieving ID code could offer better privacy (by relative information gain, as for code \mathcal{C}_2 in Table 1), for the PPRL problem, the linkage quality and implementation complexity (especially in terms of computational cost and efficiency) are important to address to make it salable and attractive for the application. This work hopefully will inspire the information theorists to have a fresh look at the identification problem in a potential application scenario (where the code is not necessarily optimal in capacity-achieving) and bring new formulations and possible solutions to the PPRL problems and other domains in practice.

Acknowledgment

The main part of the work was conducted as the author was with the University of Duisburg-Essen and was supported by the German Research Foundation under the research grant DFG 407023611. The author would like to acknowledge the fruitful discussions with Prof. Rainer Schnell, Prof. Frederik Armknecht, and Youthe Heng on PPRL; and with Prof. Dr. Han Vinck, Dr. Christian Deppe and Caspar Von Lengerke on ID codes.

References

- [ONS (2013)] Office for national statistics: Beyond 2011: Matching anonymous data, 2013.
- [Ahlsweede and Dueck (1989)] R. Ahlsweede and G. Dueck. Identification via channels. *IEEE Transactions on Information Theory*, 35(1):15–29, 1989.
- [Ahlsweede and Zhang (1995)] R. Ahlsweede and Z. Zhang. New directions in the theory of identification via channels. *IEEE Transactions on Information Theory*, 41(4):1040–1050, 1995.
- [Ariel et al. (2015)] A. Ariel, B. Bakker, M. Groot, G. Grootheest, J. Laan, J. Smit, and B. Verkerk. *Record linkage in health data: a simulation study*. Statistics Netherlands, 2015. ISBN 978903571786-6.
- [Becker and Jenkin (2015)] G. Becker and B. Jenkin. fastdigest: Fast, low memory-footprint digests of R objects. <https://cran.r-project.org/src/contrib/Archive/fastdigest/>. R package version 0.6-3, 2015.
- [Chen (2020)] Y. Chen. Current approaches and challenges for the two-party privacy-preserving record linkage (pprl). *CODASSCA 2020 - Proceedings of the 2020 International Workshop on Collaborative Technologies and Data Science in Smart City Applications*, Sep. 14-17, 2020.
- [Chen (2024)] Y. Chen. Application of identification codes to the two-party privacy-preserving record linkage (pprl). *CODASSCA 2024 - Proceedings of the 4th Workshop on Collaborative Technologies and Data Science in Smart City Applications*, Oct. 3-6, 2024.
- [Christen (2012)] P. Christen. *Data Matching - Concepts and Techniques for Record Linkage, Entity Resolution, and Duplicate Detection*. Springer, Berlin, 2012.
- [Christen and Verykios (2012)] P. Christen and V. Verykios. A tutorial on privacy-preserving record linkage. https://www.academia.edu/75300246/A_Tutorial_on_Privacy_Preserving_Record_Linkage. 2012.
- [Christen et al. (2020)] P. Christen, T. Ranbaduge, and R. Schnell. *Linking Sensitive Data - Methods and Techniques for Practical Privacy-Preserving Information Sharing*. Springer, Berlin, 2020.
- [GDPR] Council of the European Union. Council Regulation (EU) no 679/2016, 2016. <https://gdpr-info.eu/>.
- [den Boer (1993)] B. den Boer. A simple and key-economical unconditional authentication scheme. *Journal of Computer Security*, 2:65–71, 1993.
- [Derebeyoğlu et al. (2020)] S. Derebeyoğlu, C. Deppe, and R. Ferrara. Performance analysis of identification codes. *Entropy*, 22(10), 2020.
- [Han and Verdú (1992)] T. Han and S. Verdú. New results in the theory of identification via channels. *IEEE Transactions on Information Theory*, 38(1):14–25, 1992.
- [Hefele et al. (2022)] A. Hefele, C. von Lengerke, R. Ferrara, J. A. Cabrera, and F. H. P. Fitzek. ID-Sim: A simulation framework for message identification. <https://gitlab.com/alexander.hefele/id-simulator>, 2022.
- [Kurosawa and Yoshida (1999)] K. Kurosawa and T. Yoshida. Strongly universal hashing and identification codes via channels. *IEEE Transactions on Information Theory*, 45(6):2091–2095, 1999.
- [Lengerke et al. (2023)] C. V. Lengerke, A. Hefele, J. A. Cabrera, O. Kosut, M. Reisslein, and F. H. P. Fitzek. Identification codes: A topical review with design guidelines for practical systems. *IEEE Access*, 11:14961–14982, 2023.
- [Nitz and Mandal (2024)] L. Nitz and A. Mandal. Bloom encodings in DGA detection: Improving machine learning privacy by building on privacy-preserving record linkage. *JUCS - Journal of Universal Computer Science*, 30(9):1224–1243, 2024.

[Schnell et al. (2009)] R. Schnell, T. Bachteler, and J. Reiher. Privacy-preserving record linkage using Bloom filters. *BMC Medical Informatics & Decision Making*, 9:41, 2009.

[Shannon (1948)] C. E. Shannon. A mathematical theory of communication. *Bell Syst. Tech. J.*, 27(3):379–423, 1948.

[Stinson (1994)] D. Stinson. Universal hashing and authentication codes. *Designs, Codes and Cryptography*, 4:369–380, 1994.

[Vatsalan and Christen (2014)] D. Vatsalan and P. Christen. Scalable privacy-preserving record linkage for multiple databases. *CIKM 2014 - Proceedings of the 2014 ACM International Conference on Information and Knowledge Management*, Nov. 2014.

[Verdú and Wei (1993)] S. Verdú and V. Wei. Explicit construction of optimal constant-weight codes for identification via channels. *IEEE Transactions on Information Theory*, 39(1):30–36, 1993.