

Privacy-Preserving Database Union

Andrew Wildenberg
awildenberg@cornellcollege.edu
Department of Computer Science
Cornell College

— *with* —

Alberto Maria Segre
segre@cs.uiowa.edu
Department of Computer Science
The University of Iowa

Veronica Vieland
vielandv@pediatrics.ohio-state.edu
Columbus Children's Research Institute
Ohio State University

Ying Zhang
yizha@math.uiowa.edu
Applied Mathematics
The University of Iowa

The Problem

Multi-site clinical studies rely on centralized analysis of data collected from multiple locations.

Thus, centralized data analysis entails computing the set-theoretic union of multiple data sets.

However, federal regulations (*e.g.*, HIPPA) prohibit the sharing of identifying information.

Without identifying information, duplicate records cannot be found and removed.

Why We Care

The problem arises in *genetic linkage analysis* studies, where computationally-intensive statistical methods are used to find likely locations for a disease's underlying gene(s).

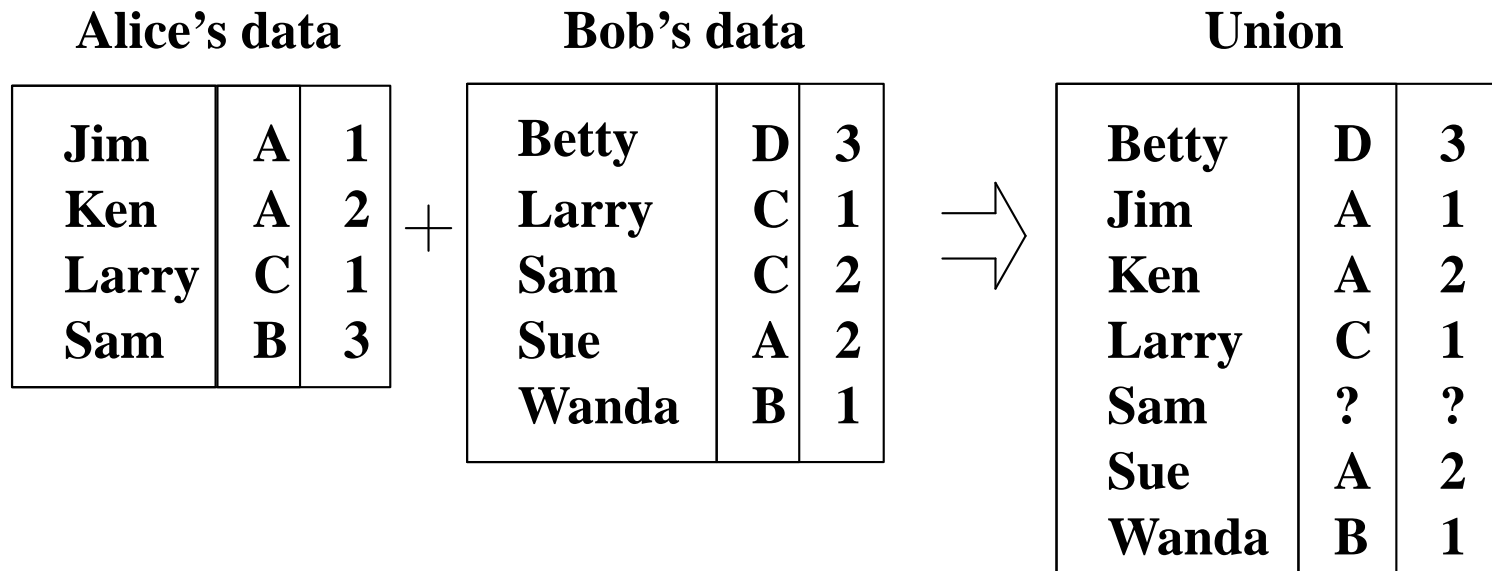
Typically, we are studying (rare) genetic disorders, with unsystematic recruitment of affected families.

Motivated families often enroll in more than one study.

But our statistical analyses assume each record is independent; simple concatenation of multiple data sets can distort the results obtained.

An Example

Alice owns $A = \{I_a, D_a\}_i$ for $0 \leq i < |A|$, and Bob owns $B = \{I_b, D_b\}_j$ for $0 \leq j < |B|$, where I_a and I_b are *identifier fields* and D_a and D_b are *data fields*, wherein no candidate key exists.



We expect that the data fields D_a and D_b , at least, will be subject to noise or measurement error.

A Trusted Third Party Solution

Well-designed multi-site clinical studies will plan ahead for centralized data analysis, asking each subject to consent to sharing of data with a central site.

Some data handling protocols may even call for use of one-way hash functions to blind the identifiers before transferring records to the central site.

The central site acts as a *trusted third party*, and can properly compute the set-theoretic union of the data sets, resolving differences in matching data fields according to some pre-established criteria.

But what happens when subjects have not consented *a priori*? Or when data collected previously might subsequently be reused? Or when no trusted third party is available?

Privacy-Preserving Data Mining

Privacy-preserving data mining is the term used to describe research on data mining over privately-held data sets.

Prior work in this area is of two types:

- (1) *Distortion techniques*, also called *data camouflaging*, rely on disclosing appropriately perturbed versions of only the data fields (so that the recipient cannot later link data to identities). But proper camouflaging requires detailed knowledge of the intended analysis algorithm, and does not remove duplicate records.
- (2) *Secure multiparty computation* allows two or more parties to collaboratively compute the desired result (some less pure solutions permit exchange of intermediate results as well). But SMC is cumbersome, expensive, and does not directly address the duplicate record problem.

Kantarcioglu and Clifton (2004)

In their work on distributed data mining of association rules, Kantarcioglu and Clifton (2004) propose a partial solution based on the use of a *commutative cipher*, where $K_a K_b(M) = K_b K_a(M)$ for $C = K(M)$ and $M = K^{-1}(C)$.

However, their solution requires the records be revealed in the last step of the protocol: it only protects information about who originally owned the record, and not information (*e.g.*, identifying information) that is part of the record itself.

Furthermore, their protocol is only suitable for three or more parties, as, in the two party case, what didn't belong to Alice must have come from Bob.

In this talk, we'll negotiate the set-theoretic union after the fact, without violating privacy concerns, and without resorting to a trusted third party.

Assumptions

We assume communication between parties is secure, such as might occur over an encrypted channel.

We assume all parties have properly authenticated: everyone is in fact exactly who they claim to be.

We assume all parties are basically honest and cooperative, but still curious (this is usually called a *semihonest model*).

We assume a commutative cipher (*e.g.*, Pohlig-Hellman) is available.

We assume availability of a keyed commutative one-way hash function, where $H(M)$ denotes the postimage of hash H applied to message M and $H_1 H_2(M) = H_2 H_1(M)$.

Our solution is a protocol that describes a fixed set of message exchanges between Alice and Bob that result in Alice computing the set-theoretic union of the two data sets.

Initial State

Alice knows:

Jim	A 1
Ken	A 2
Larry	C 1
Sam	B 3

Bob knows:

Betty	D 3
Larry	C 1
Sam	C 2
Sue	A 2
Wanda	B 1

(note divergence in Sam's record)

Step 1

Alice provides Bob with copies of her records, where the identifier fields are hashed using a one-way keyed hash function H_a and the data fields are encrypted using a commutative cipher with random key K_a , both of Alice's choosing.

$$\text{Alice} \rightarrow \text{Bob}: \quad \{H_a(I_a), K_a(D_a)\}_i \quad 0 \leq i < |A|$$

The hash key H_a and cipher key K_a are Alice's secrets, and should never be revealed.

After Step 1

Alice knows:

Jim	A 1
Ken	A 2
Larry	C 1
Sam	B 3

Bob knows:

Betty	D 3
Larry	C 1
Sam	C 2
Sue	A 2
Wanda	B 1

Jim	A 1
Ken	A 2
Larry	C 1
Sam	B 3

(shading represents encryption)

Step 2

Bob hashes copies of Alice's previously hashed identifiers using a one-way keyed hash H_b of his choosing and shuffles the result, returning the doubly-hashed identifiers to Alice in some random order.

$$\text{Bob} \rightarrow \text{Alice: } \{H_b H_a(I_a)\}_i \quad 0 \leq i < |A|$$

He then retains a copy of Alice's doubly-hashed identifiers and their associated encrypted data fields “in escrow” for later use.

The hash key H_b is Bob's secret, and should never be revealed.

After Step 2

Alice knows:

Jim	A 1	Jim
Ken	A 2	Ken
Larry	C 1	Larry
Sam	B 3	Sam

Bob knows:

Betty	D 3
Larry	C 1
Sam	C 2
Sue	A 2
Wanda	B 1

Jim	A 1
Ken	A 2
Larry	C 1
Sam	B 3

(shuffles not shown)

Step 3

Bob next provides Alice with copies of his own records, where the identifiers are hashed using the same one-way keyed hash function H_b of the previous step, and the data fields are encrypted using a random key K_b of Bob's own choosing.

$$\text{Bob} \rightarrow \text{Alice:} \quad \{H_b(I_b), K_b(D_b)\}_j \quad 0 \leq j < |B|$$

The key K_b is Bob's secret, and should never be revealed.

After Step 3

Alice knows:

Jim	A 1
Ken	A 2
Larry	C 1
Sam	B 3

Jim	
Ken	
Larry	
Sam	

Betty	D 3
Larry	C 1
Sam	C 2
Sue	A 2
Wanda	B 1

Bob knows:

Betty	D 3
Larry	C 1
Sam	C 2
Sue	A 2
Wanda	B 1

Jim	A 1
Ken	A 2
Larry	C 1
Sam	B 3

(shuffles not shown)

Step 4

Alice applies her one-way keyed hash H_a to Bob's hashed identifiers and the commutative cipher using her key K_a to Bob's encrypted data fields. At this point, Alice knows both

$$\begin{aligned} \text{Alice: } & \{H_b H_a(I_a)\}_i & 0 \leq i < |A| \\ & \{H_a H_b(I_b), K_a K_b(D_b)\}_j & 0 \leq j < |B| \end{aligned}$$

and recall that, because we are using a commutative one-way hash and a commutative cipher, $H_b H_a(x) = H_a H_b(x)$ and $K_b K_a(x) = K_a K_b(x)$.

After Step 4

Alice knows:

Jim	A 1
Ken	A 2
Larry	C 1
Sam	B 3

Jim	
Ken	
Larry	
Sam	

Betty	D 3
Larry	C 1
Sam	C 2
Sue	A 2
Wanda	B 1

Betty	D 3
Larry	C 1
Sam	C 2
Sue	A 2
Wanda	B 1

Bob knows:

Betty	D 3
Larry	C 1
Sam	C 2
Sue	A 2
Wanda	B 1

Jim	A 1
Ken	A 2
Larry	C 1
Sam	B 3

(shuffles not shown)

Step 5

Alice now computes the union of the doubly-hashed identifiers and their doubly-encrypted associated data, if any, and then fills out the missing data fields with random bit strings, R . Each R must be identical in size to the encrypted data fields, so that all the records in the union have like format and size. She then shuffles the result, and returns it to Bob in some random order.

$$\text{Alice} \rightarrow \text{Bob}: \quad \{H_a H_b(I_{ab}), \{K_a K_b(D_b) \vee R\}\}_l \quad 0 \leq l < |U_{ab}|$$

where $I_{ab} = I_a \cup I_b$ and each doubly-hashed identifier $H_a H_b(I_{ab})$ is paired with either its doubly-encrypted data field $K_a K_b(D_b)$, if available, or else some random pattern R .

After Step 5

Alice knows:

Jim	A 1
Ken	A 2
Larry	C 1
Sam	B 3

Jim	
Ken	
Larry	
Sam	

Betty	D 3
Larry	C 1
Sam	C 2
Sue	A 2
Wanda	B 1

Betty	D 3
Larry	C 1
Sam	C 2
Sue	A 2
Wanda	B 1

Betty	D 3
Jim	
Ken	
Larry	C 1
Sam	C 2
Sue	A 2
Wanda	B 1

Bob knows:

Betty	D 3
Larry	C 1
Sam	C 2
Sue	A 2
Wanda	B 1

Jim	A 1
Ken	A 2
Larry	C 1
Sam	B 3

Betty	D 3
Jim	
Ken	
Larry	C 1
Sam	C 2
Sue	A 2
Wanda	B 1

(shuffles not shown; random fillers *R* shown as blanks)

Step 6

Bob decrypts, using K_b , the doubly-encrypted data fields. Since the random fillers R are just random bit sequences, $K_b^{-1}(R)$ is also a random bit sequence, and remains indistinguishable, to Bob, from $K_b^{-1}K_bK_a(D_b) = K_a(D_b)$. Next, Bob reassociates Alice's original data fields (which he had previously, in step 2, retained "in escrow") with the appropriately hashed identifiers, overwriting their current data fields, producing:

$$\text{Bob: } \{H_a H_b(I_{ab}), K_a(D_{ab})\}_l \quad 0 \leq l < |U_{ab}|$$

where D_{ab} denotes the data attributes associated each record in I_{ab} with D_a taking precedence over D_b for records in the intersection of the two data sets. Note that the overwritten fields are either just random bit strings or encrypted, and therefore unrecognizable, versions of Bob's own data for records that appear in both data sets.

After Step 6

Alice knows:

Jim	A 1
Ken	A 2
Larry	C 1
Sam	B 3

Jim	
Ken	
Larry	
Sam	

Betty	D 3
Larry	C 1
Sam	C 2
Sue	A 2
Wanda	B 1

Betty	D 3
Larry	C 1
Sam	C 2
Sue	A 2
Wanda	B 1

Betty	D 3
Jim	
Ken	
Larry	C 1
Sam	C 2
Sue	A 2
Wanda	B 1

Bob knows:

Betty	D 3
Larry	C 1
Sam	C 2
Sue	A 2
Wanda	B 1

Betty	D 3
Jim	A 1
Ken	A 2
Larry	C 1
Sam	B 3
Sue	A 2
Wanda	B 1

Jim	A 1
Ken	A 2
Larry	C 1
Sam	B 3

Betty	D 3
Jim	
Ken	
Larry	C 1
Sam	C 2
Sue	A 2
Wanda	B 1

(shuffles not shown; note replacement of Sam's data fields with escrowed versions)

Step 7

Bob strips the identifiers, shuffles the results, and returns the data fields to Alice.

$$\text{Bob} \rightarrow \text{Alice: } \{K_a(D_{ab})\}_l \quad 0 \leq l < |U_{ab}|$$

After Step 7

Alice knows:

Jim	A 1
Ken	A 2
Larry	C 1
Sam	B 3

Jim	
Ken	
Larry	
Sam	

Betty	D 3
Larry	C 1
Sam	C 2
Sue	A 2
Wanda	B 1

Betty	D 3
Larry	C 1
Sam	C 2
Sue	A 2
Wanda	B 1

Betty	D 3
Jim	
Ken	
Larry	C 1
Sam	C 2
Sue	A 2
Wanda	B 1

D 3
A 1
A 2
C 1
B 3
A 2
B 1

Bob knows:

Betty	D 3
Larry	C 1
Sam	C 2
Sue	A 2
Wanda	B 1

Betty	D 3
Jim	A 1
Ken	A 2
Larry	C 1
Sam	B 3
Sue	A 2
Wanda	B 1

Jim	A 1
Ken	A 2
Larry	C 1
Sam	B 3

Betty	D 3
Jim	
Ken	
Larry	C 1
Sam	C 2
Sue	A 2
Wanda	B 1

D 3
A 1
A 2
C 1
B 3
A 2
B 1

(shuffles not shown)

Step 8

Alice decrypts, using K_a , the data fields received from Bob, to produce records consisting of the now unblinded, data fields:

$$\text{Alice: } \{D_{ab}\}_l \quad 0 \leq l < |U_{ab}|$$

thereby obtaining the set-theoretic union of the two original data sets.

D 3
A 1
A 2
C 1
B 3
A 2
B 1

(shuffles not shown; note Alice sees only her own version of Sam's data)

Analysis

Both Alice and Bob learn $|U_{ab}|$, the size of the union. This is hard to avoid (although it is easy for Alice to "pad" her data and therefore obscure the true size of the intersection from Bob; it is also feasible, although slightly more difficult, for Bob to do the same).

No I_a or I_b is ever revealed; both Alice and Bob see encrypted versions of each other's identifiers, but these are as secure as the ciphers.

Assume that Alice would like to guess which of her patients are in fact in the intersection of the two datasets. Known plaintext attacks are extremely difficult, and compounded by the fact that, thanks to shuffling, ciphertext/plaintext pairs are nearly impossible to identify.

Indeed, the chance of Alice guessing the correct mapping for q of the $H_b(I_a)$ hashes in $A \cap B$ to q of her original I_a plaintexts is 1 in $\frac{|A|!}{(|A| - q)!}$, which is vanishingly small for any interesting values of q and $|A|$.

Analysis (continued)

Moreover, while Alice and Bob can both compute $|A \cap B| = |A| + |B| - |U_{ab}|$, the size of the intersection, neither learns *which* of their own records are in the intersection.

More precisely:

$$\text{Prob}(\{I_b, D_b\} \in A \cap B) = \frac{|A \cap B|}{|B|} = \frac{|A| + |B| - |U_{ab}|}{|B|}$$

which goes to 1 when $|U_{ab}| = |A|$ and goes to 0 when $|U_{ab}| = |A| + |B|$ (a symmetric formulation holds for $\text{Prob}(\{I_a, D_a\} \in A \cap B)$).

So in the special case where all (or none) of one participant's records are replicated in the other participant's data set, the extent of the overlap is clear once $|U_{ab}|$, $|A|$, and $|B|$ are known. In other cases, random guessing about whether a particular record is in the intersection of the data sets is the best anyone can hope to do.

Analysis (continued)

Alice computes the true set-theoretic union (modulo noise) without obtaining or divulging any identifying subject information (note Bob can execute the symmetric protocol to get his own version of the union).

Solution works for horizontally partitioned data (this example) as well as for vertically partitioned or mixed data.

Since each participant encrypts/decrypts each record at most once with each of at most two keys, the cost of the two-party protocol is $O(n)$ where $n = |A| + |B|$

Protocol has been extended to three or more parties.

Conclusion

A general solution to the privacy-preserving database union problem.

Identifiers remain encrypted and secure.

No third party required.

No information leakage; participants do not learn identities of records in the intersection.

Has been extended to three or more parties.

Efficient; unlike SMC, computing (encryption) and communication costs are essentially linear in the number of records.

Prototype implemented in Java.

Acknowledgements

For their helpful comments and feedback:

Robert Hanson Ramon Lawrence
Meredith Patterson Alessio Signorini

Support for this research was provided in part by the National Science Foundation through grants ITR/ACI0218491 (AMS) and CCLI/A&I0511391 (APW), the National Alliance for Autism Research through the Autism Genome Project grant (VJV), and the National Institutes of Health through grant R01/NS042165 (VJV).