PRIVACY PRESERVING FUZZY PATIENT MATCHING USING HOMOMORPHIC ENCRYPTION

ETELEMED 2020

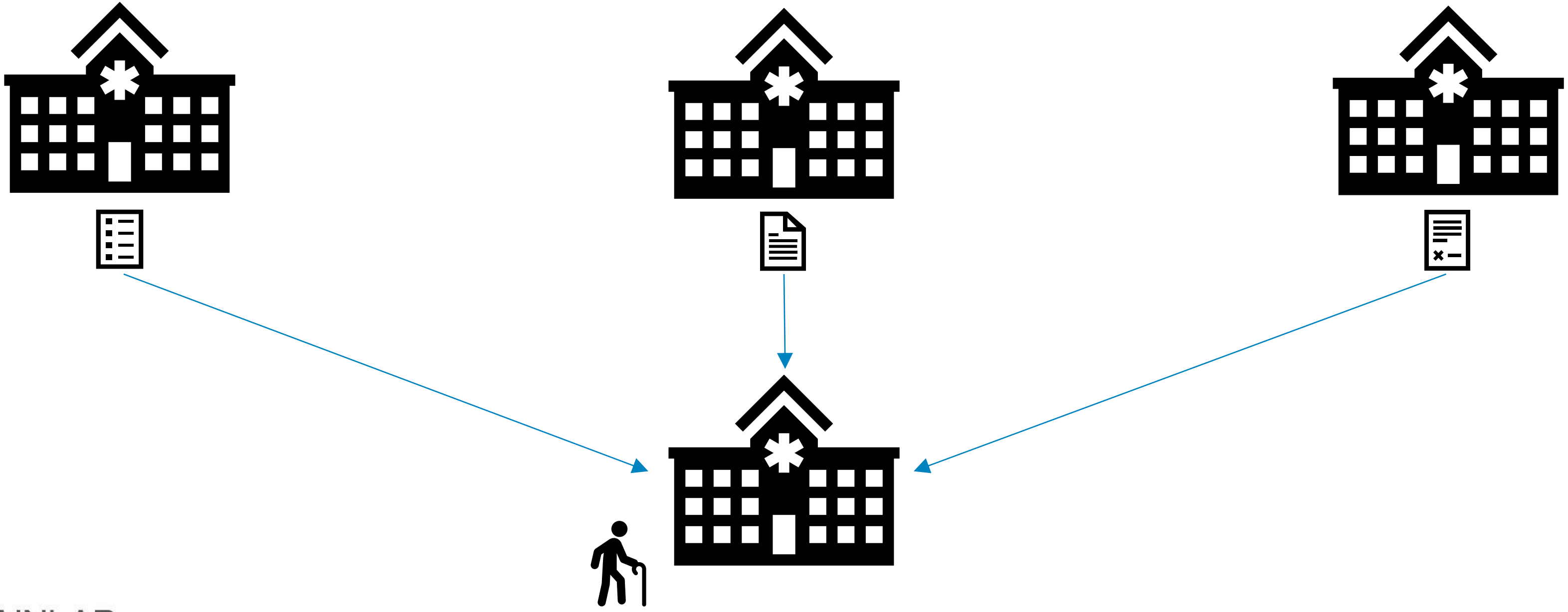
Shiva Ashish Thumparthy

November 7, 2020

OBJECTIVE

Medical record interoperability

- Consider a patient’s longitudinal medical history
- Provide better patient outcomes and higher quality of service



CHALLENGES

No universal identifier for linkage

Quasi-identifiers such as name, birthday and recent address are most-often used

- Cannot be shared across facilities or with third parties

Cannot rely on literal matches due to errors in demographics

EXISTING SOLUTIONS

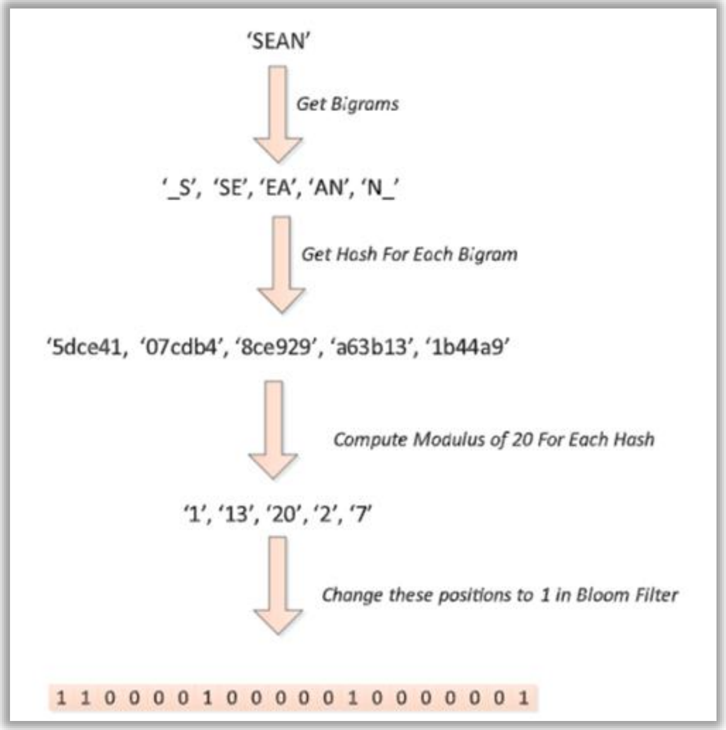
Bloom filter[1] based

Data structure to obtain digests of information without revealing original data

Makes use of multiple hash functions to mask inputs

Digests can be compared to arrive at similarities between two Bloom filters

Privacy preserving(?)



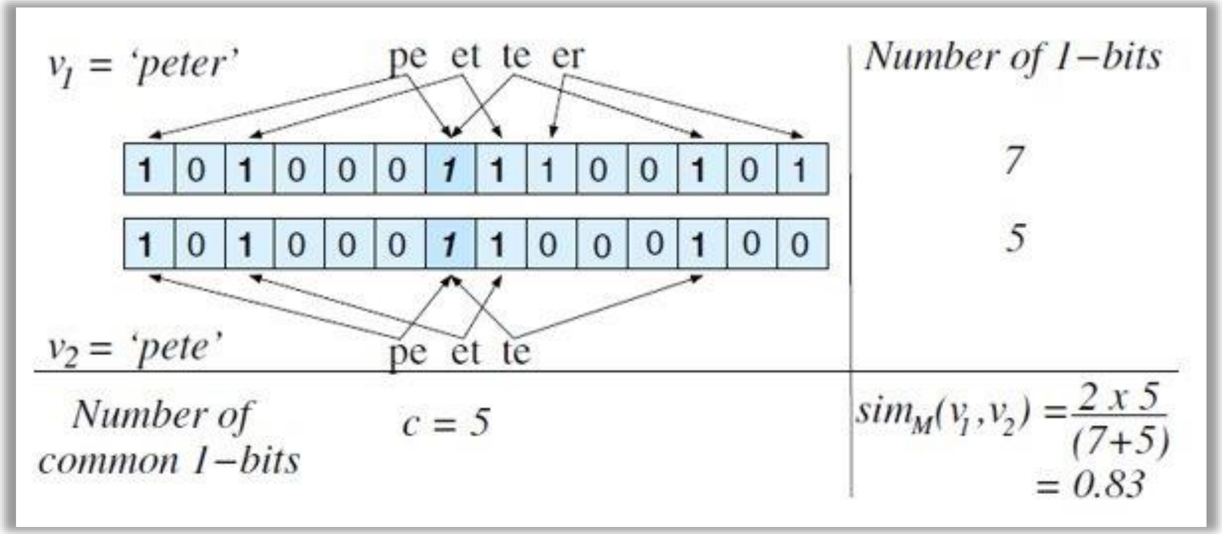
EXISTING SOLUTIONS

Calculating similarity between Bloom filters[2]

Intuitively, number of 1-bits in same positions (common 1-bits) vs total number of 1-bits (total 1-bits)

Predefined threshold for match

Example 1: Dice coefficient[2] = (2 * common 1-bits) / total 1-bits



EXISTING SOLUTIONS

Calculating similarity between Bloom filters

Example 2: Threshold Tversky index[3] = $(\theta_n + \theta_d)$ * common 1-bits – θ_n * total 1-bits

- Reveals only binary result, rather than similarity score
- Does not require division

BF_1 =

1	1	1	1	1

BF_2 =

1	1	0	1	0

BF_3 =

1	1	0	1	1

Threshold(θ) = 80% i.e. $\theta = 8/10 = 4/5$ i.e. $\theta_n = 4$, $\theta_d = 5$

Tversky(BF_1, BF_2)

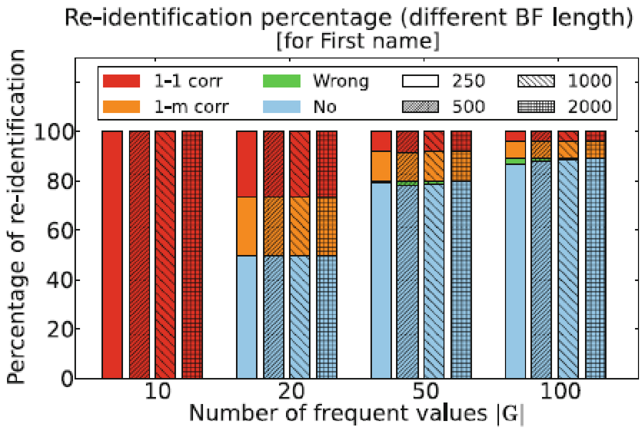
- Common 1-bits = 3; Total 1-bits = 8
- Result = 9(3) - 4(8) = 27 – 32 = -5 (Mismatch)

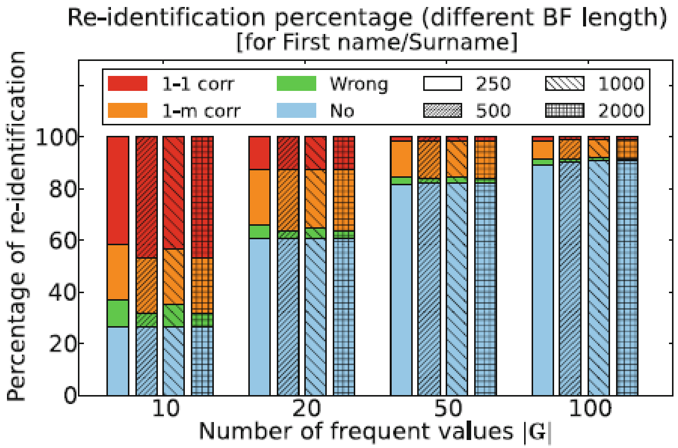
Tversky(BF_1, BF_3)

- Common 1-bits = 4; Total 1-bits = 9
- Result = 9(4) - 4(9) = 36 – 36 = 0 (Match)

ISSUES WITH EXISTING SOLUTIONS

Frequency and cryptanalysis attacks, brute force attacks[4]

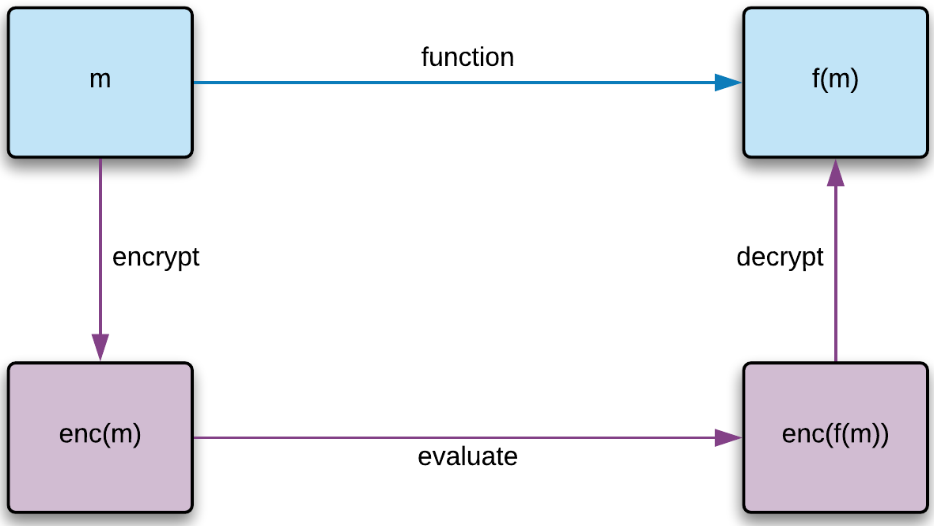




HOMOMORPHIC ENCRYPTION

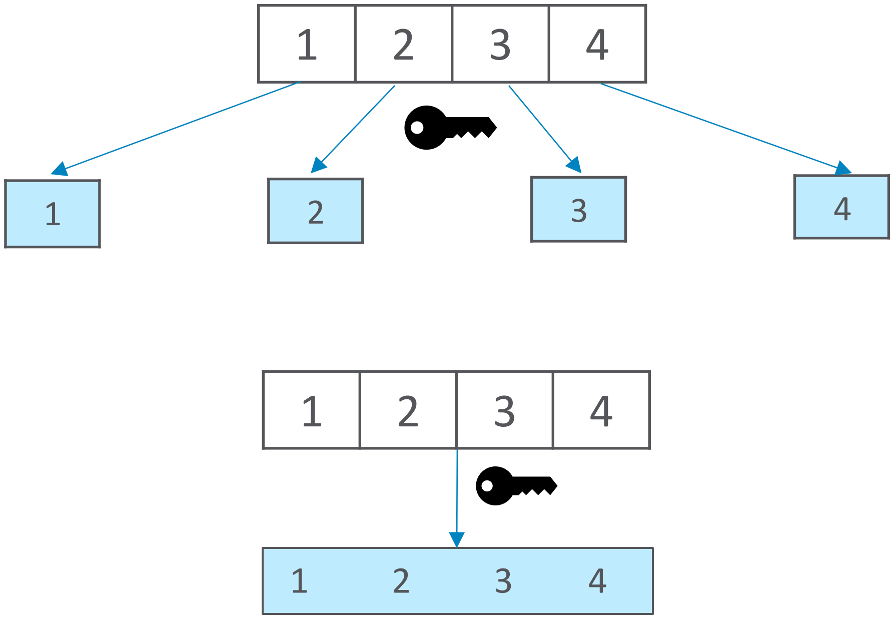
Allows computation on ciphertexts, generating an encrypted result

Result, when decrypted, matches the result of the operations as if they had been performed on the plaintext



HOMOMORPHIC ENCRYPTION

Ciphertext packing of vectors



HOMOMORPHIC ENCRYPTION

Ciphertext packing of vectors

Encryption of multiple values into one ciphertext, as opposed to a single value

Embed values of vectors into coefficients of polynomials

1	2	3	4

$$P(x) = 1x^3 + 2x^2 + 3x^1 + 4x^0$$

1	1	0	0

$$Q(x) = 1x^3 + 1x^2 + 0x^1 + 0x^0$$

HOMOMORPHIC ENCRYPTION

Inner products

One vector needs to be inverted i.e. reversed

The result of the inner product is the coefficient of $x^{length-1}$

1	2	3	4

Map → $P(x) = x^3 + 2x^2 + 3x + 4$ Enc → $P(x) = x^3 + 2x^2 + 3x + 4$

1	2	3	4

Inv + Map → $P(x) = 4x^3 + 3x^2 + 2x + 1$ Enc → $P(x) = 4x^3 + 3x^2 + 2x + 1$

$(x^3 + 2x^2 + 3x + 4) * (4x^3 + 3x^2 + 2x + 1) = 4x^6 + 11x^5 + 20x^4 + \mathbf{30x^3} + 20x^2 + 11x + 4$

Inner product = Coefficient of $x^{length-1}$ = Coefficient of x^{4-1} = Coefficient of x^3 = **30**

VECTOR-BASED MATCHING SOLUTION

Encrypt the bits of Bloom filters using homomorphic encryption[5][6]

Compare encrypted Bloom filters

- Does not reveal any information to third parties

Results can only be decrypted by the intended recipient

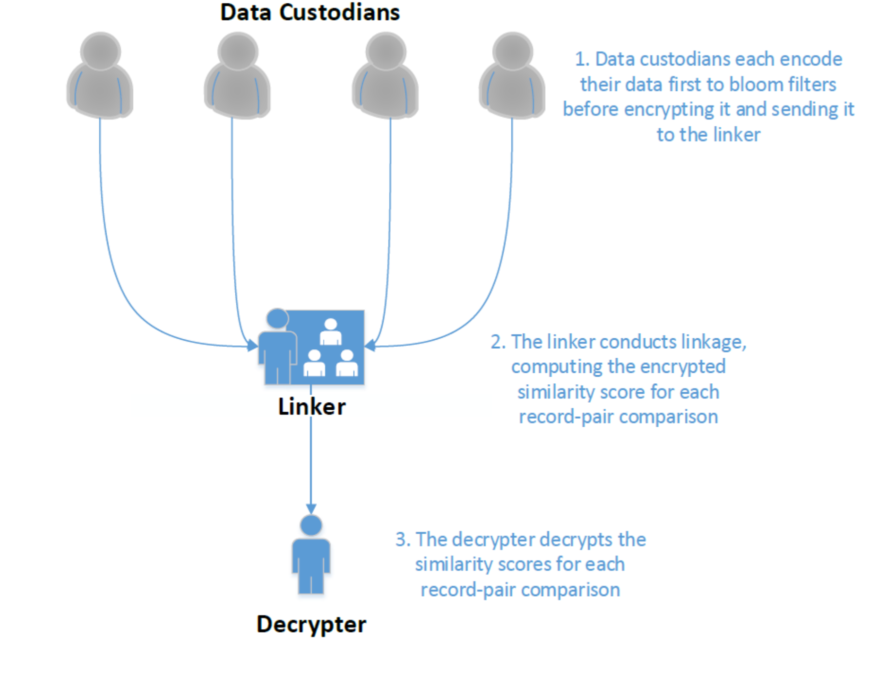
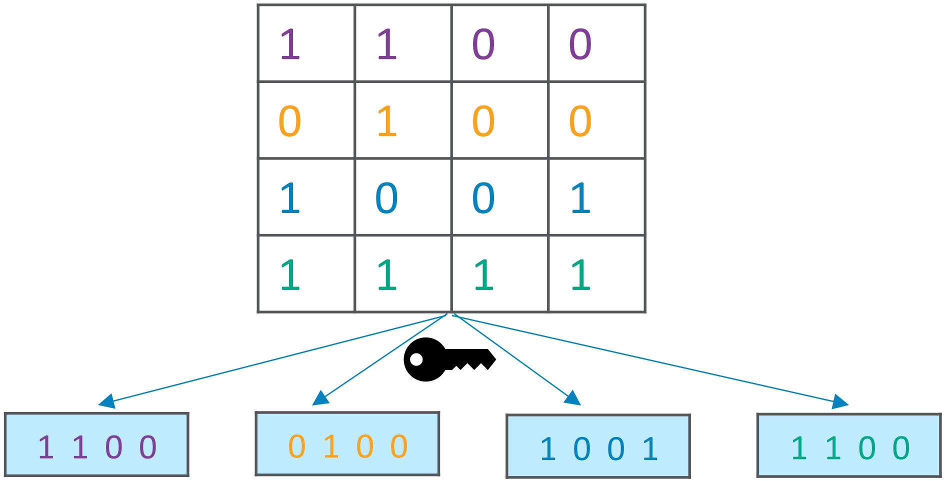
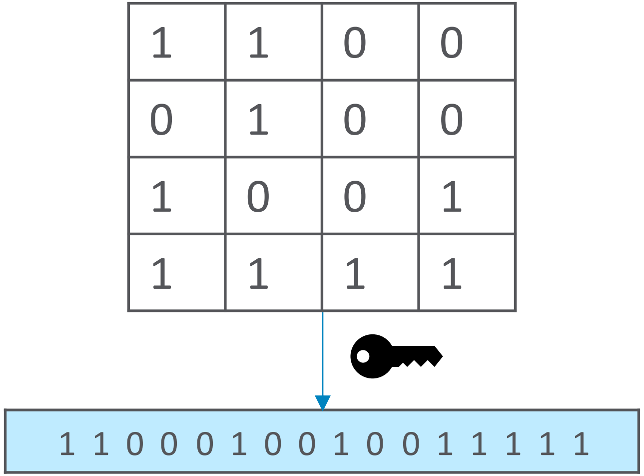


Figure 1: Data movements for the proposed protocol

HOMOMORPHIC ENCRYPTION
Ciphertext packing of matrices





HOMOMORPHIC ENCRYPTION

Ciphertext packing of matrices

Matrices can be packed into one ciphertext[7]

- Intuition: rows are packed as per vector packing, then combined into a single polynomial

Facility A	Bit1	Bit2	Bit3	Bit4	Bit5
A1	1	0	0	1	0
A2	0	1	1	1	0
A3	0	1	0	1	1
A4	1	1	0	0	1
A5	0	0	0	0	1

Bloom bits	Bit1	Bit2	Bit3	Bit4	Bit5	Bit6	Bit7	Bit8	Bit9	Bit10	Bit11	Bit12	Bit13	Bit14	Bit15	Bit16	Bit17	Bit18	Bit19	Bit20	Bit21	Bit22	Bit23	Bit24	Bit25
Values	1	0	0	1	0	0	1	1	1	0	0	1	0	1	1	1	1	0	0	1	0	0	0	0	1

$$P(x) = x^{24} + x^{21} + x^{18} + x^{17} + x^{16} + x^{13} + x^{11} + x^{10} + x^{9} + x^{8} + x^{5} + 1$$

PROPOSED SOLUTION

Matrix multiplication

Multiplication gives number of 1-bits in the same location (common 1-bits) for each pair of records.

Facility A	Bit1	Bit2	Bit3	Bit4	Bit5
A1	1	0	0	1	0
A2	0	1	1	1	0
A3	0	1	0	1	1

Multiply

Facility B'	B1	B2	B3	B4	B5
Bit1	0	1	0	1	1
Bit2	1	0	0	0	1
Bit3	0	0	1	0	0
Bit4	1	1	1	0	1
Bit5	1	0	1	1	0

Result A*B'	B1	B2	B3	B4	B5
A1	1	2	1	1	2
A2	2	1	2	0	2
A3	3	1	2	1	2

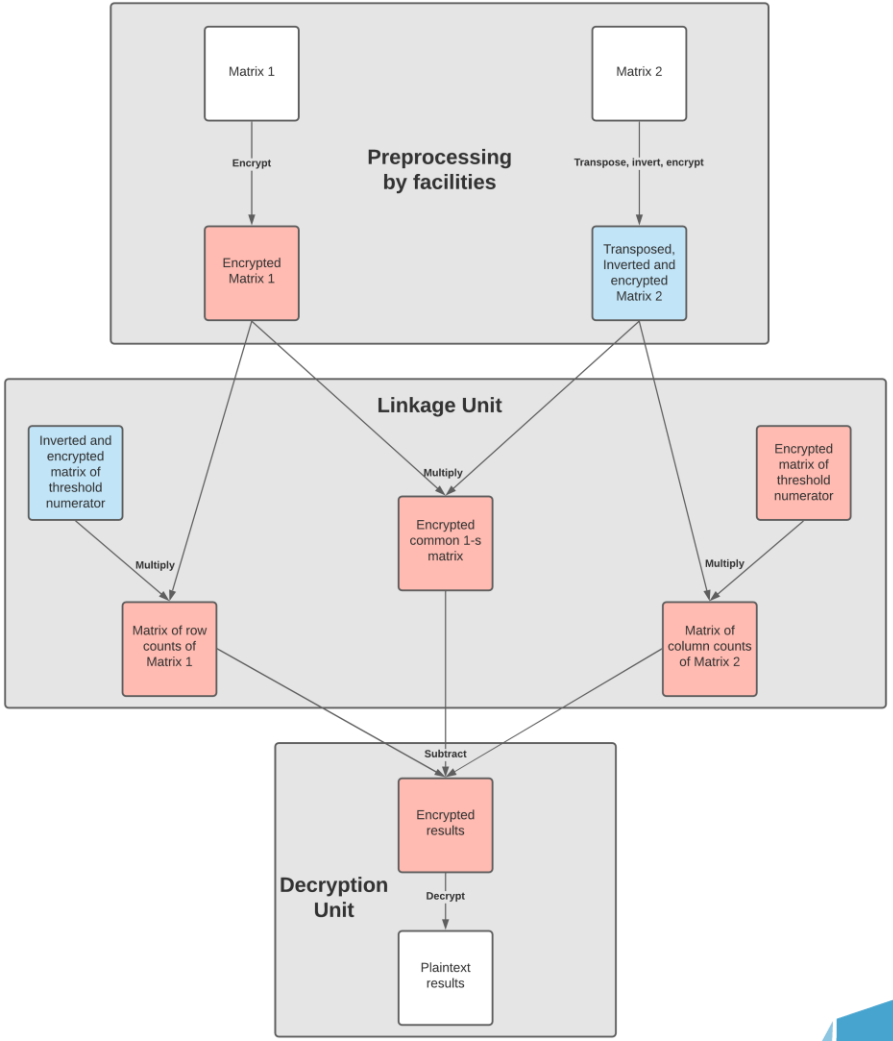
PROPOSED SOLUTION

Bloom filters are stacked i.e. treated as rows of a matrix

Bloom matrix * transpose of another Bloom matrix = pairwise common 1-bits

Bloom matrix * matrix of all 1s = 1-bits count of that matrix

Threshold Tversky index calculated from these 3 matrices



PRELIMINARY RESULTS

Matrix size	Time taken(s)			
	Vector encryption	Matrix encryption	Vector matching	Matrix Matching
4*4	0.0626682	0.0482091	2.07659	0.058472
8*8	0.125355	0.0412167	8.06762	0.056312
16*16	0.252727	0.10382	32.1595	0.147115
32*32	0.502159	4.01244	128.199	5.84446

OPEN PROBLEMS

Large key size of homomorphic encryption keys

- Key size of the order of ~1Gb required to encrypt 32 * 32 matrices

Multiplication of large matrices is very computationally intensive

- Can be fixed using bootstrapping
    - Intuition: manages noise in ciphertext by encrypting again

REFERENCES

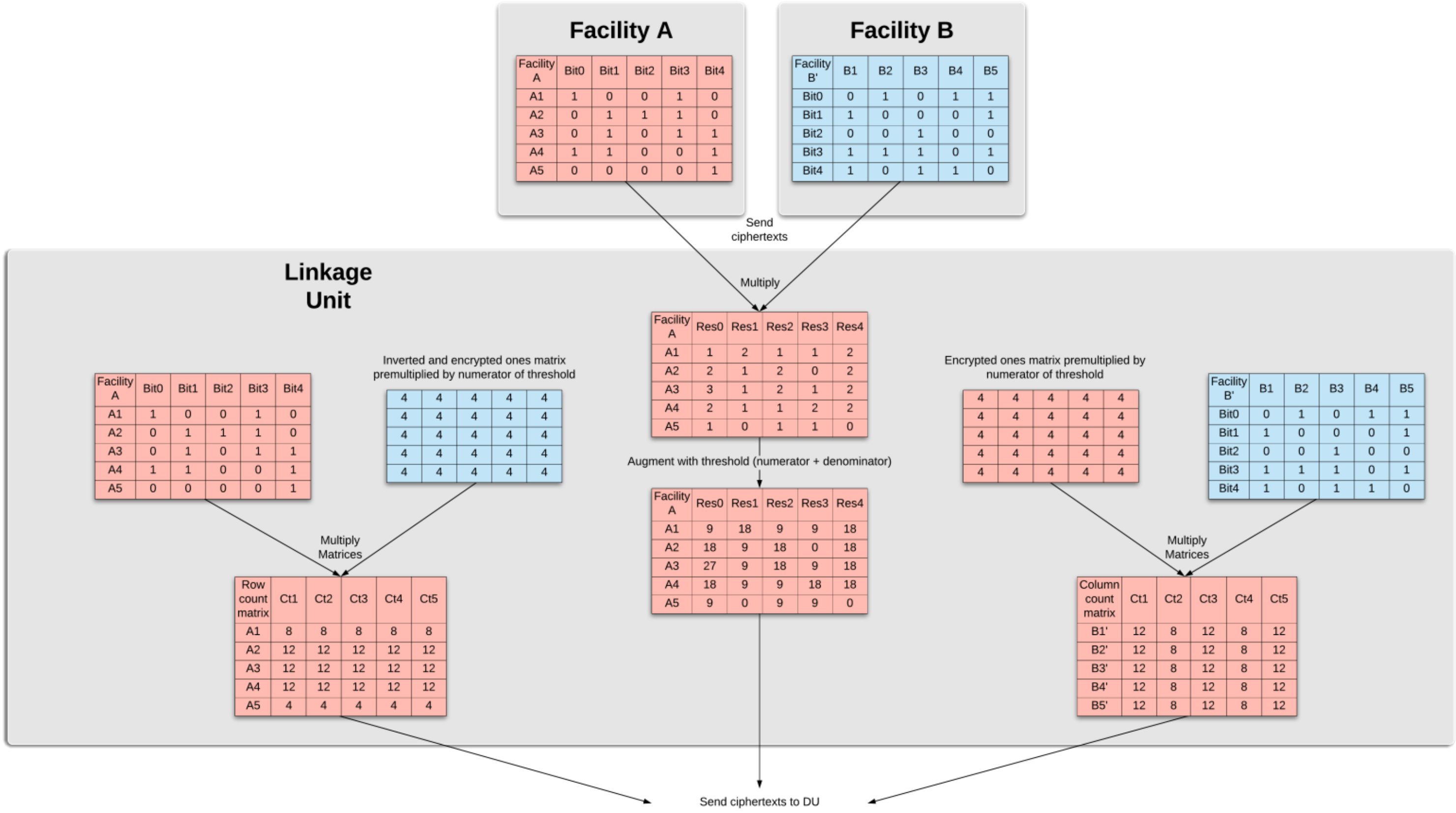
1. R. Schnell, T. Bachteler. and J. Reiher, "Privacy-preserving record linkage using Bloom filters", BMC medical informatics and decision making, 2009, *9*(1), p.41.
2. D. Vatsalan, & P. Christen, Privacy-preserving matching of similar patients, "Journal of biomedical informatics", 2016, *59*, pp. 285-298.
3. K. Shimizu, K. Nuida, H. Arai, S. Mitsunari, N. Attrapadung, M. Hamada, K. Tsuda, T. Hirokawa, J. Sakuma, G. Hanaoka and K. Asai, "Privacy-preserving search for chemical compound databases", BMC bioinformatics 16, 2015, no. S18, S6.
4. P. Christen, R. Schnell, D. Vatsalan, and T. Ranbaduge, "Efficient cryptanalysis of bloom filters for privacy-preserving record linkage", Pacific-Asia Conference on Knowledge Discovery and Data Mining, Springer, Cham, May 2017, pp. 628-640.
5. S. M. Randall, A. P. Brown, A. M. Ferrante, J. H. Boyd, and J. B. Semmens, “Privacy preserving record linkage using homomorphic encryption”, Population Informatics for Big Data, Aug.2015, 10.
6. M. S. H. Cruz, T. Amagasa, C. Watanabe, W. Lu, and H. Kitagawa, "Secure similarity joins using fully homomorphic encryption", Proceedings of the 19th International Conference on Information Integration and Web-based Applications & Services, ACM, Dec.2017, pp 224-233.
7. D. H. Duong, P. K. Mishra, and M. Yasuda, “Efficient secure matrix multiplication over LWE-based homomorphic encryption”, Tatra Mountains mathematical publications, 67(1), Sep. 2016, pp. 69-83.

THANK YOU

Shiva Ashish Thumparthy
Brainlab AG
ashish.thumparthy@brainlab.com

EXAMPLE

Linkage Unit



EXAMPLE
Decryption Unit

Decryption Unit

Decrypt row count vector and generate row count matrix

Row count matrix	Ct1	Ct2	Ct3	Ct4	Ct5
A1	8	8	8	8	8
A2	12	12	12	12	12
A3	12	12	12	12	12
A4	12	12	12	12	12
A5	4	4	4	4	4

Decrypt multiplied matrix

Facility A	Res0	Res1	Res2	Res3	Res4
A1	9	18	9	9	18
A2	18	9	18	0	18
A3	27	9	18	9	18
A4	18	9	9	18	18
A5	9	0	9	9	0

Decrypt column count vector and generate column count matrix

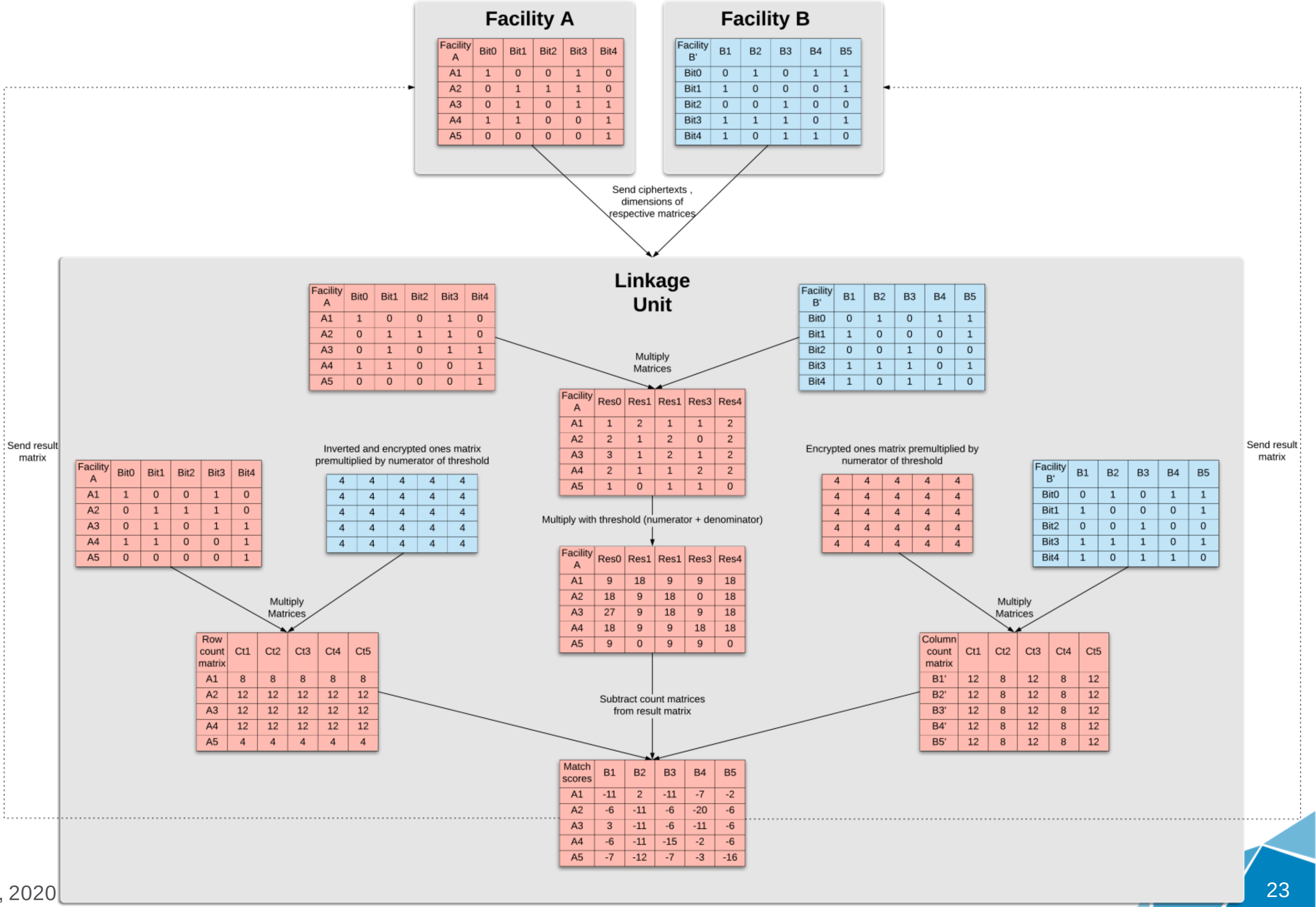
Column count matrix	Ct1	Ct2	Ct3	Ct4	Ct5
B1'	12	8	12	8	12
B2'	12	8	12	8	12
B3'	12	8	12	8	12
B4'	12	8	12	8	12
B5'	12	8	12	8	12

Subtract row and column matrix from multiplied matrix

Match scores	B1	B2	B3	B4	B5
A1	-11	2	-11	-7	-2
A2	-6	-11	-6	-20	-6
A3	3	-11	-6	-11	-6
A4	-6	-11	-15	-2	-6
A5	-7	-12	-7	-3	-16

EXAMPLE

Single
Linkage
Party



Facility A	Bit0	Bit1	Bit2	Bit3	Bit4
A1	1	0	0	1	0
A2	0	1	1	1	0
A3	0	1	0	1	1
A4	1	1	0	0	1
A5	0	0	0	0	1

Facility B'	B1	B2	B3	B4	B5
Bit0	0	1	0	1	1
Bit1	1	0	0	0	1
Bit2	0	0	1	0	0
Bit3	1	1	1	0	1
Bit4	1	0	1	1	0

Facility A	Bit0	Bit1	Bit2	Bit3	Bit4
A1	1	0	0	1	0
A2	0	1	1	1	0
A3	0	1	0	1	1
A4	1	1	0	0	1
A5	0	0	0	0	1

Facility B'	B1	B2	B3	B4	B5
Bit0	0	1	0	1	1
Bit1	1	0	0	0	1
Bit2	0	0	1	0	0
Bit3	1	1	1	0	1
Bit4	1	0	1	1	0

Facility A	Res0	Res1	Res1	Res3	Res4
A1	1	2	1	1	2
A2	2	1	2	0	2
A3	3	1	2	1	2
A4	2	1	1	2	2
A5	1	0	1	1	0

Facility A	Bit0	Bit1	Bit2	Bit3	Bit4
A1	1	0	0	1	0
A2	0	1	1	1	0
A3	0	1	0	1	1
A4	1	1	0	0	1
A5	0	0	0	0	1

4	4	4	4	4
4	4	4	4	4
4	4	4	4	4
4	4	4	4	4
4	4	4	4	4

4	4	4	4	4
4	4	4	4	4
4	4	4	4	4
4	4	4	4	4
4	4	4	4	4

Facility B'	B1	B2	B3	B4	B5
Bit0	0	1	0	1	1
Bit1	1	0	0	0	1
Bit2	0	0	1	0	0
Bit3	1	1	1	0	1
Bit4	1	0	1	1	0

Facility A	Res0	Res1	Res1	Res3	Res4
A1	9	18	9	9	18
A2	18	9	18	0	18
A3	27	9	18	9	18
A4	18	9	9	18	18
A5	9	0	9	9	0

Row count matrix	Ct1	Ct2	Ct3	Ct4	Ct5
A1	8	8	8	8	8
A2	12	12	12	12	12
A3	12	12	12	12	12
A4	12	12	12	12	12
A5	4	4	4	4	4

Column count matrix	Ct1	Ct2	Ct3	Ct4	Ct5
B1'	12	8	12	8	12
B2'	12	8	12	8	12
B3'	12	8	12	8	12
B4'	12	8	12	8	12
B5'	12	8	12	8	12

Match scores	B1	B2	B3	B4	B5
A1	-11	2	-11	-7	-2
A2	-6	-11	-6	-20	-6
A3	3	-11	-6	-11	-6
A4	-6	-11	-15	-2	-6
A5	-7	-12	-7	-3	-16