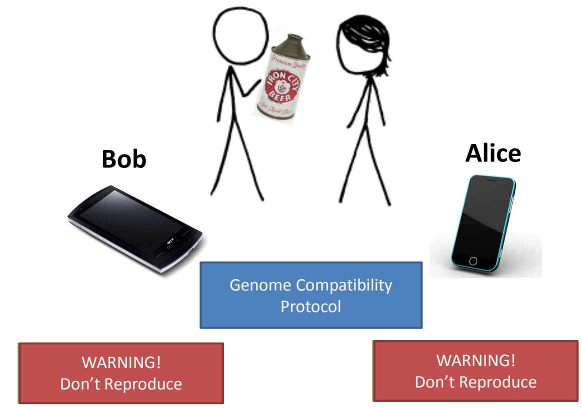
Secure Computation in the Real(ish) World

Carnegie Mellon
20 April 2011

David Evans
University of Virginia
http://www.cs.virginia.edu/evans
http://www.MightBeEvil.com

"Genetic Dating"



GenePartner™ Love is no coincidence

23andMe

TheScientist News Current Issue Archive Sur

2 comments
Comment on this news story
By Kerry Grens

Forget mistletoe - what about DNA?

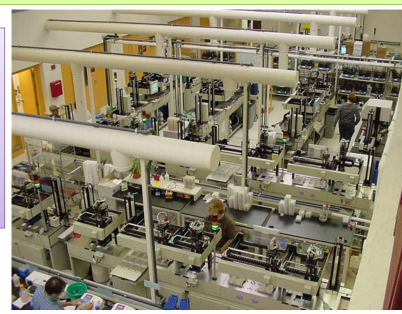
A new dating service matches singles using major histocompatibility complex genes

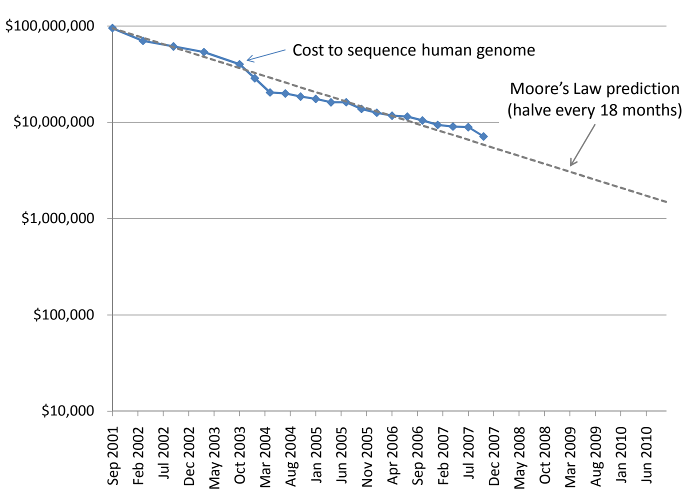
ScientificMatch.com "The Science of Love"

Genome Sequencing

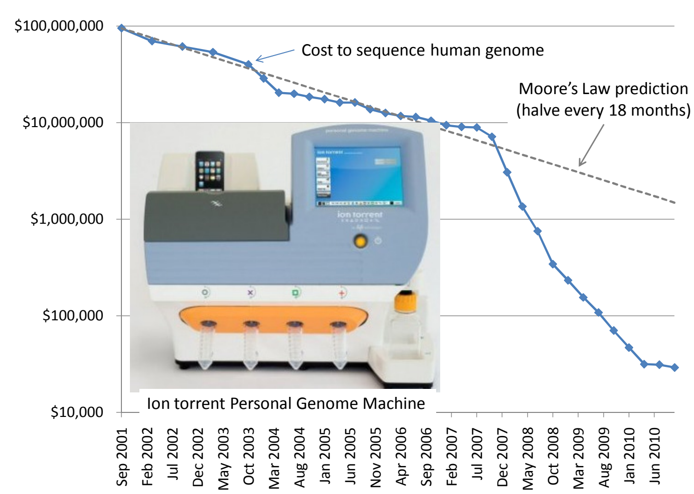
1990: Human Genome Project starts, estimate $3B to sequence one genome ($0.50/base)

2000: Human Genome Project declared complete, cost ~$300M


Whitehead Institute, MIT



Data from National Human Genome Research Institute: http://www.genome.gov/sequencingcosts



Data from National Human Genome Research Institute: http://www.genome.gov/sequencingcosts

Year	r	Reported sequencing consumables cost	Estimated cost per 40-fold coverage
[redacted]		$10,000,000	$57,000,000
		$1,000,000	$5,700,000
		$250,000	$330,000
		$48,000	$69,000
2009	t	$8,005	$3,700
2009	t	$3,451	$2,200
2009	t	$1,726	$1,500

George Church (Personal Genome Project)

Human Geno ... ***ling DNA Nanoarrays*.** Radoje Drmanac, An... L. Burns, Bahram G. Kermani, Paolo Carnevali, Igo... Andres Fernandez, Bryan Staker, Krishna P. Par... Ryan Cedeno, Linsu Chen, Dan Chernikoff, Alex... t, Coleen R. Hacker, Robert Hartlage, Brian Hauser, S... lvin Kong, Tom Landers, Catherine Le, Jia Liu, Celeste E. McBride, Matt Morenzoni, Robert E. Morey, Karl Mutch, Helena Perazich, Kimberly Perry, Brock A. Peters, Joe Peterson, Charit L. Pethiyagoda, Kaliprasad Pothuraju, Claudia Richter, Abraham M. Rosenbaum, Shaunak Roy, Jay Shafto, Uladzislau Sharanhovich, Karen W. Shannon, Conrad G. Sheppy, Michel Sun, Joseph V. Thakuria, Anne Tran, Dylan Vu, Alexander Wait Zaranek, Xiaodi Wu, Snezana Drmanac, Arnold R. Oliphant, William C. Banyai, Bruce Martin, Dennis G. Ballinger, George M. Church, Clifford A. Reid. ***Science*, January 2010.**

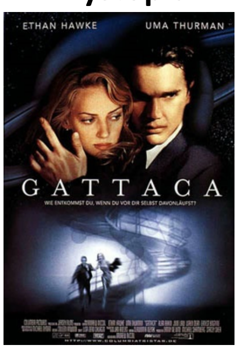
Steven Pinker (PGP-10)

LETTERS

On Jim Watson's *APOE* status: genetic information is hard to hide

European Journal of Human Genetics (2009) **17**, 147–149; doi:10.1038/ejhg.2008.198; published online 22 October 2008

Dystopia



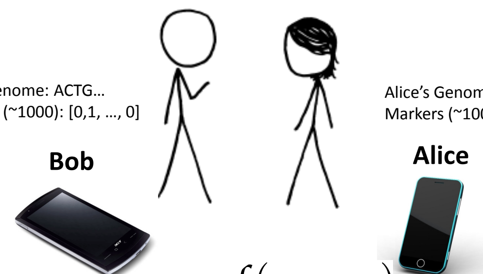
Gene Partner™ Love is no coincidence

Personalized Medicine

Secure Two-Party Computation

Bob's Genome: ACTG…
Markers (~1000): [0,1, …, 0]

Alice's Genome: ACTG…
Markers (~1000): [0, 0, …, 1]



$$x = f(g_A, g_B)$$

Can Alice and Bob compute a function of their private data, without exposing anything about their data besides the result?

Secure Function Evaluation

Alice (circuit generator) Picks $a \in \{0,1\}^s$

Agree on $f(a,b) \to x$

Bob (circuit evaluator) Picks $b \in \{0,1\}^t$

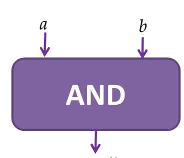
Garbled Circuit Protocol

Outputs $x = f(a,b)$ without revealing a to Bob or b to Alice.

Andrew Yao, 1982/1986

Yao's Garbled Circuits

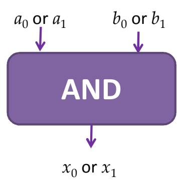
Inputs		Output
a	b	x
0	0	0
0	1	0
1	0	0
1	1	1



Computing with Meaningless Values?

Inputs		Output
a	b	x
a_0	b_0	x_0
a_0	b_1	x_0
a_1	b_0	x_0
a_1	b_1	x_1

a_i, b_i, x_i are **random** values, chosen by the **circuit generator** but **meaningless** to the **circuit evaluator**.

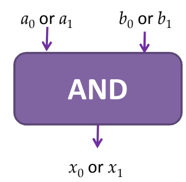


Computing with Garbled Tables

Inputs		Output
a	b	x
a_0	b_0	$Enc_{a_0,b_0}(x_0)$
a_0	b_1	$Enc_{a_0,b_1}(x_0)$
a_1	b_0	$Enc_{a_1,b_0}(x_0)$
a_1	b_1	$Enc_{a_1,b_1}(x_1)$

Bob can only decrypt **one** of these!

a_i, b_i, x_i are **random** values, chosen by the **circuit generator** but **meaningless** to the **circuit evaluator**.



Garbled And Gate
$Enc_{a_0, b_1}(x_0)$
$Enc_{a_1,b_1}(x_1)$
$Enc_{a_1,b_0}(x_0)$
$Enc_{a_0,b_0}(x_0)$

Garbled Circuit Protocol

Alice (circuit generator)

Creates random keys: $a_0, a_1, b_0, b_1, x_0, x_1$

Bob (circuit evaluator)

And Gate
$Enc_{a_0, b_1}(x_0)$
$Enc_{a_1,b_1}(x_1)$
$Enc_{a_1,b_0}(x_0)$
$Enc_{a_0,b_0}(x_0)$

Sends a_i to Bob based on her input value

a_0

How does the Bob learn his own input wires?

Primitive: **Oblivious Transfer**

Alice

Knows b_0, b_1

Bob

Picks $i \in \{0, 1\}$

Oblivious Transfer Protocol

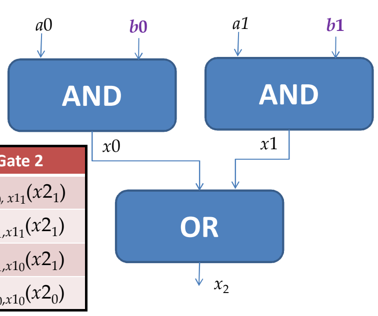
Learns nothing

Learns b_i (only)

Oblivious: Alice doesn't learn which secret Bob obtains
Transfer: Bob learns one of Alice's secrets

Rabin, 1981; Even, Goldreich, and Lempel, 1985; many subsequent papers

Chaining Garbled Circuits



And Gate 1
$Enc_{a10, b11}(x1_0)$
$Enc_{a11,b11}(x1_1)$
$Enc_{a11,b10}(x1_0)$
$Enc_{a10,b10}(x1_0)$

Or Gate 2
$Enc_{x00, x11}(x2_1)$
$Enc_{x01,x11}(x2_1)$
$Enc_{x01,x10}(x2_1)$
$Enc_{x00,x10}(x2_0)$

We can do ***any*** computation privately this way!

Threat Model

Semi-Honest (*Honest But Curious*) **Adversary**

Adversary follows the protocol as specified (**!**)

Curious adversary tries to learn more from protocol execution transcript

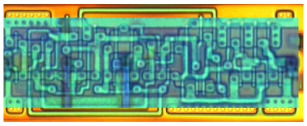
Garbled Circuits security proofs depend on this very weak model

General techniques for converting protocols secure in semi-honest model to resist malicious adversary.

Amount of information that could leak is probably small

Possibility to use software attestation to validate executing code?

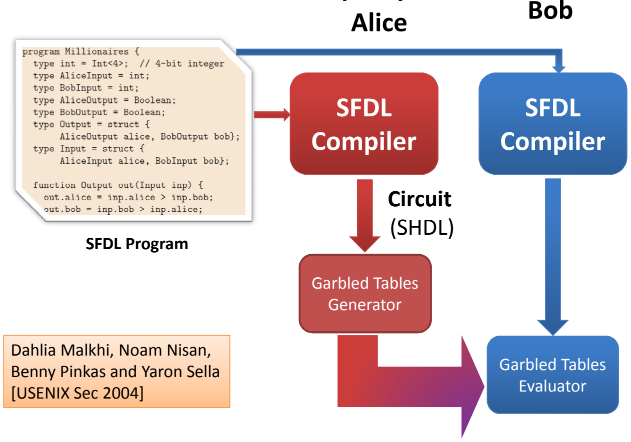
Building Computing Systems



$Enc_{x00,\,x11}(x2_1)$
$Enc_{x01,x11}(x2_1)$
$Enc_{x01,x10}(x2_1)$
$Enc_{x00,x10}(x2_0)$

Digital Electronic Circuits	Garbled Circuits
Operate on **known data**	Operate on **encrypted wire labels**
One-bit logical operation requires moving a few electrons a few nanometers (hundreds of Billions per second)	One-bit logical operation requires performing (up to) 4 encryption operations (~100,000 gates per second)
Reuse is great!	Reuse is not allowed!
All basic operations have similar cost	Some logical operations "free" (XOR, NOT)

Fairplay



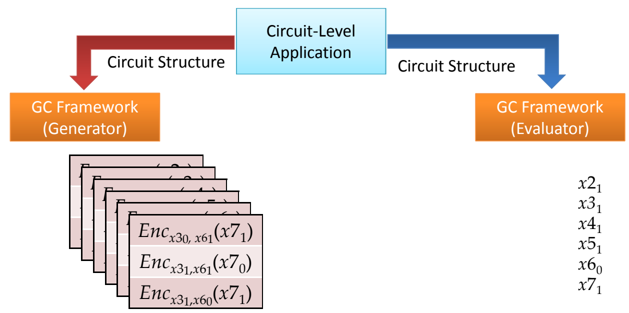
Dahlia Malkhi, Noam Nisan, Benny Pinkas and Yaron Sella [USENIX Sec 2004]

(Un)Fairplay?

An alternative approach to our protocols would have been to apply Yao's generic secure two-party protocol to the recognition algorithm. This would have required expressing the algorithm as a circuit which computes and compares many Hamming distances, and then sending and computing that circuit. … **We therefore believe that the performance of our protocols is significantly better than that of applying generic protocols.**
Margarita Osadchy, Benny Pinkas, Ayman Jarrous, Boaz Moskovich. *SCiFI – A System for Secure Face Identification*. Oakland 2010.

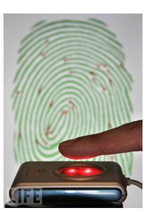
Protocol 1 (generic SMC) is very fast. Protocol 1 is ideal for small strings because the entire computation is performed in one round, but the circuit size is extremely large for longer strings. Our prototype circuit compiler can compile circuits for problems of size (200, 200) but uses almost 2 GB of memory to do so. **Significantly larger circuits would be constrained by available memory for constructing their garbled versions.**
Somesh Jha, Louis Kruger, Vitaly Shmatikov. *Towards Practical Privacy for Genomic Computation*. Oakland 2008.

Faster Garbled Circuits



Gates can be evaluated as they are generated: **pipelining**
Gates can be evaluated in any topological sort order: **parallelizing**
Garbled evaluation can be **combined with normal execution**

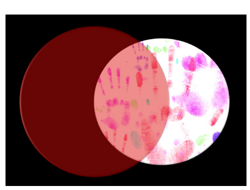
Applications



Privacy-Preserving Biometric Matching

Private Personal Genomics

Private AES Encryption



Private Set Intersection

Heterozygous Recessive Risk

Counsyl

Alice / **Bob**

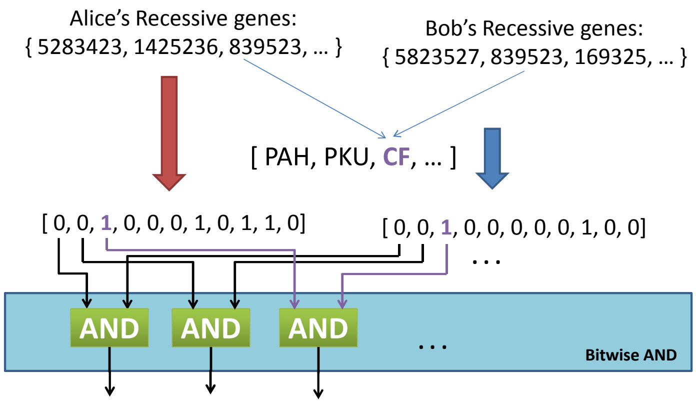
	A	a
A	AA	Aa
a	aA	aa

Aa: carrier
aa: cystic fibrosis

Alice's Heterozygous Recessive genes: { 5283423, 1425236, 839523, … }
Bob's Heterozygous Recessive genes: { 5823527, 839523, 169325, … }

Goal: find the intersection of A and B

Bit Vector Intersection



Scaling

What if there are millions of possible diseases?

Length of bit vector:

number of possible values

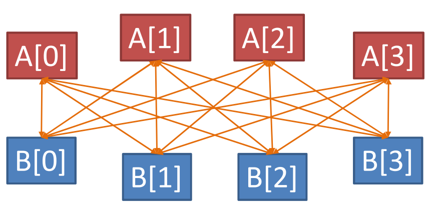
(2^L where L is number of bits for each value)

Other private set intersection problems:
- Do Alice and Bob have any friends in common?
- Data mining problems: combine medical records across hospitals
- Two companies want to do joint marketing to common customers

Pairwise Comparison

data-oblivious algorithm

```
randomly permute A
randomly permute B
for i in range(0, n-1):
   for j in range(0, n-1):
      if A[i] = B[j] output A[i]
```

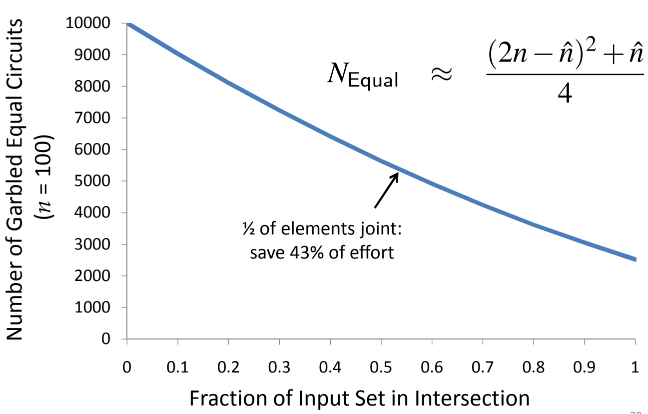


n^2 comparisons

Short-Circuit Pairwise Comparison

```
for i in range(0, n-1):
   mask[i] = false
for i in range(0, n-1):
   for j in range(0, n-1):
      if not mask[i] and A[i] = B[j]:
         reveal A[i] to both
         mask[i] = true
         break
```


Short-Circuit Analysis

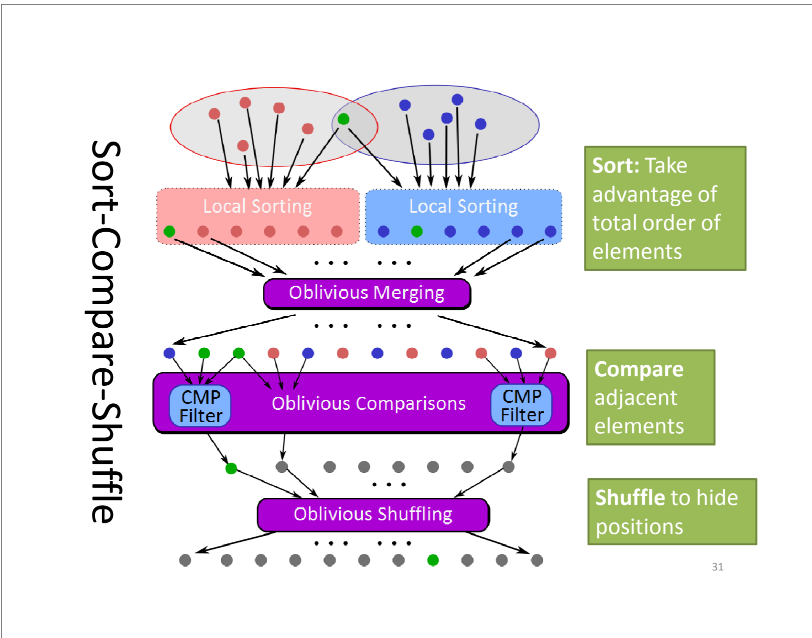


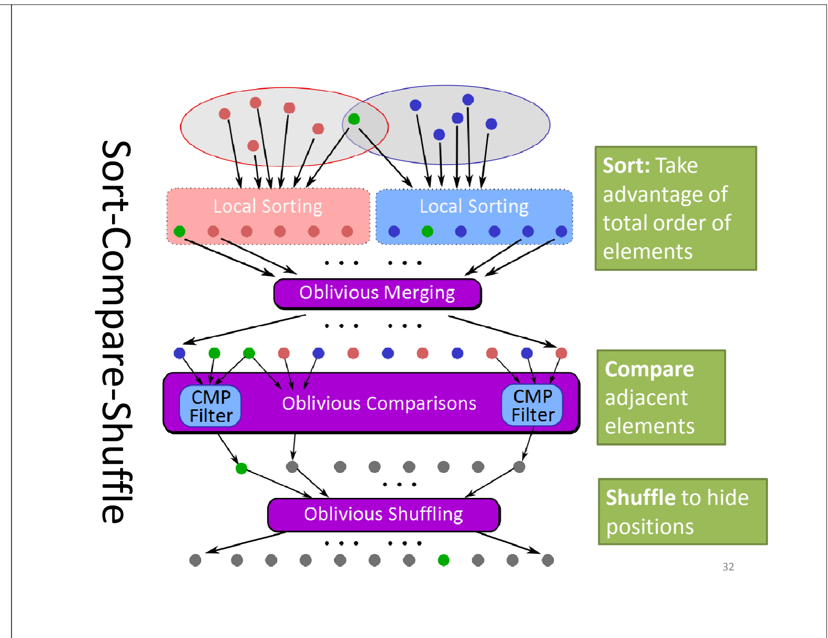
$$N_{\text{Equal}} \approx \frac{(2n-\hat{n})^2+\hat{n}}{4}$$

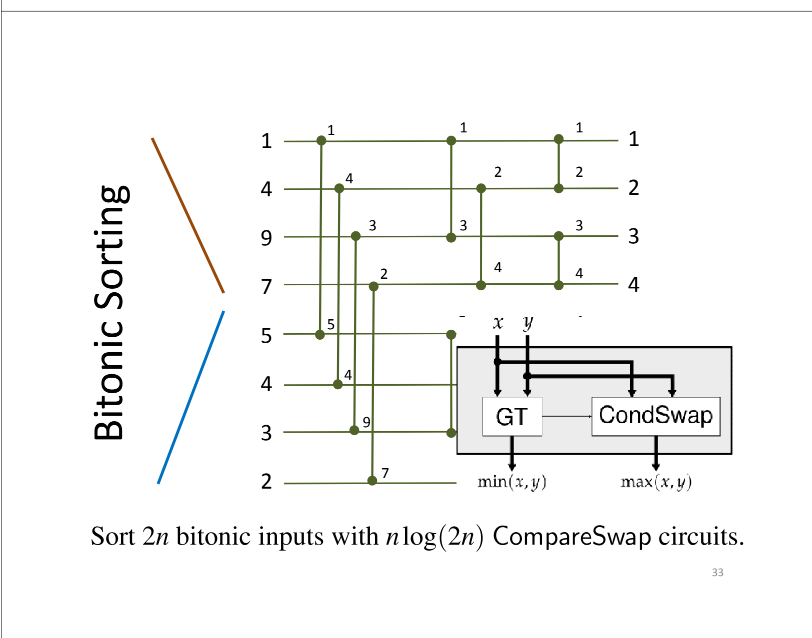
Scaling

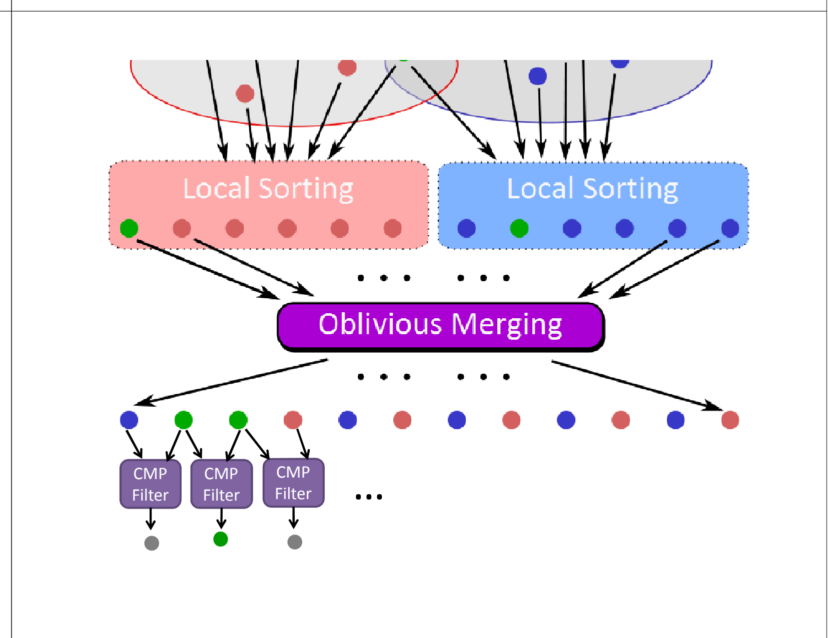
Other private set intersection problems:
- Do Alice and Bob have any friends in common?
- Data mining problems: combine medical records across hospitals
- Two companies want to do joint marketing to common customers

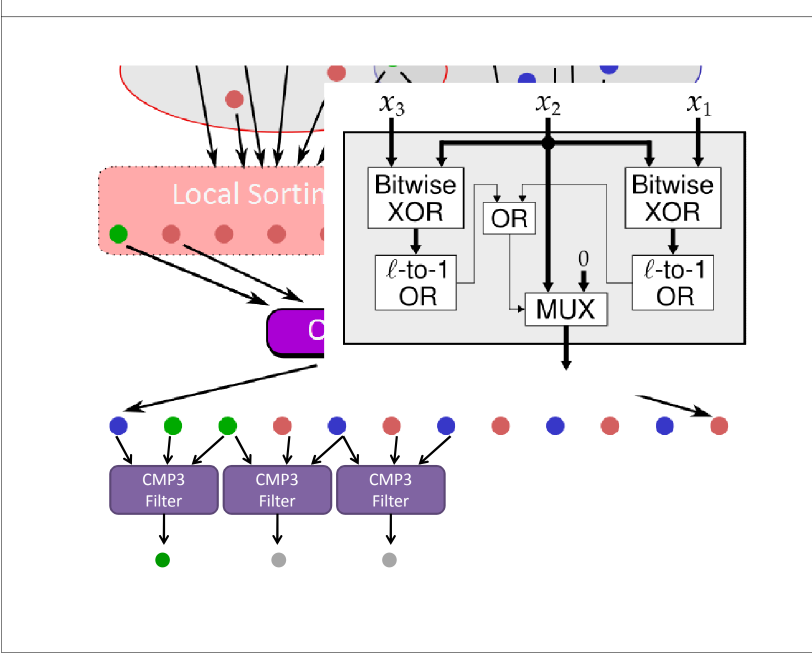
This is still $O(n^2)$. Is there an $O(n \log n)$ solution?

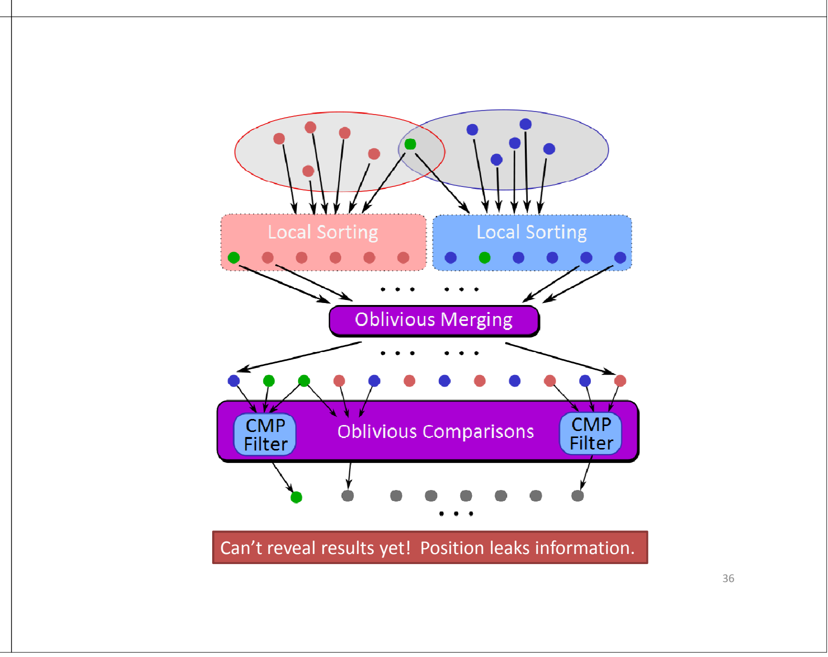

Sort-Compare-Shuffle
Local Sorting
Local Sorting
Oblivious Merging
CMP Filter
Oblivious Comparisons
CMP Filter
Oblivious Shuffling
Sort: Take advantage of total order of elements
Compare adjacent elements
Shuffle to hide positions
31


Sort-Compare-Shuffle
Local Sorting
Local Sorting
Oblivious Merging
CMP Filter
Oblivious Comparisons
CMP Filter
Oblivious Shuffling
Sort: Take advantage of total order of elements
Compare adjacent elements
Shuffle to hide positions
32


Bitonic Sorting
GT
CondSwap
min(x, y)
max(x, y)
Sort 2n bitonic inputs with n log(2n) CompareSwap circuits.
33


Local Sorting
Local Sorting
Oblivious Merging
CMP Filter
CMP Filter
CMP Filter


Local Sortin
Bitwise XOR
OR
Bitwise XOR
ℓ-to-1 OR
MUX
ℓ-to-1 OR
CMP3 Filter
CMP3 Filter
CMP3 Filter


Local Sorting
Local Sorting
Oblivious Merging
CMP Filter
Oblivious Comparisons
CMP Filter
Can't reveal results yet! Position leaks information.
36

Oblivious Shuffling

Homomorphic Encryption Shuffling Protocol

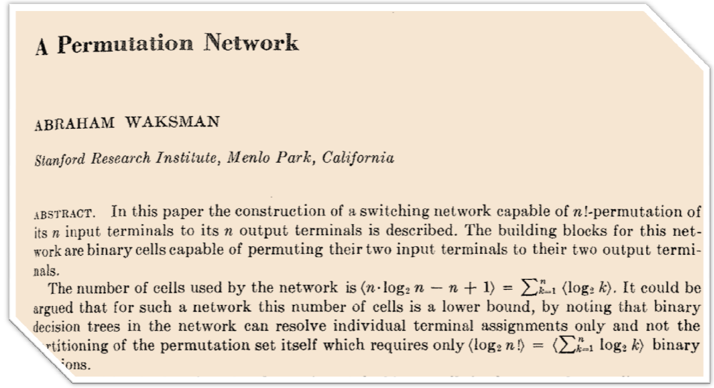
Add random mask, permute, exchange and reveal

Expensive

Sort

Simple…but expensive

Random Permutation



A Permutation Network

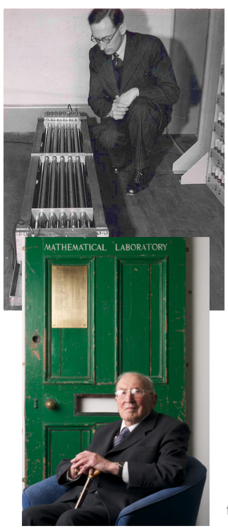
ABRAHAM WAKSMAN

Stanford Research Institute, Menlo Park, California

ABSTRACT. In this paper the construction of a switching network capable of $n!$-permutation of its n input terminals to its n output terminals is described. The building blocks for this network are binary cells capable of permuting their two input terminals to their two output terminals.

The number of cells used by the network is $\langle n \cdot \log_2 n - n + 1 \rangle = \sum_{k=1}^{n} \langle \log_2 k \rangle$. It could be argued that for such a network this number of cells is a lower bound, by noting that binary decision trees in the network can resolve individual terminal assignments only and not the rtitioning of the permutation set itself which requires only $\langle \log_2 n! \rangle = \langle \sum_{k=1}^{n} \log_2 k \rangle$ binary ions.

Journal of the ACM, January 1968

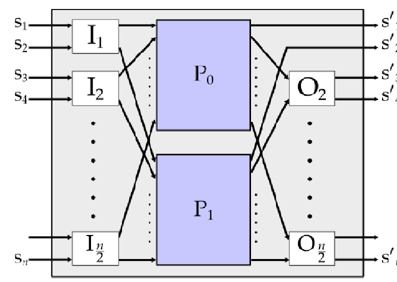


I do not imagine that many of the Turing lecturers who will follow me will be people who were acquainted with Alan Turing. … Although a mathematician, Turing took quite an interest in the engineering side of computer design… **Turing's contribution to this discussion was to advocate the use of gin, which he said contained alcohol and water in just the right proportions …**

Sir Maurice Wilkes (1913-29 Nov 2010),
Computers Then and Now
(1967 Turing Award Lecture)

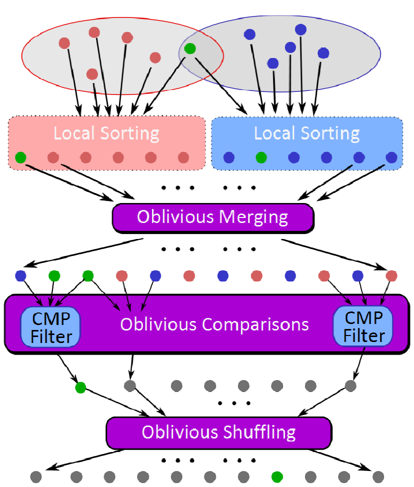
flickr: rolandeva

Waksman Network



Same circuit can generate any permutation:
select a random permutation, and pick swaps

$$n \log n - n + 1 \text{ gates}$$



Private Set ntersection Protocol

Gates to generate and evaluate

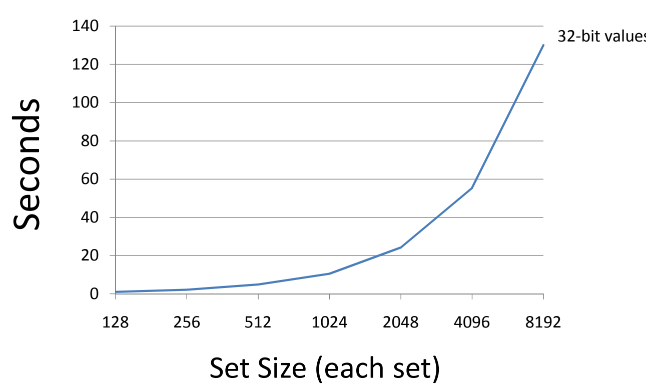
Free

$$n \log(2n) \times 2\sigma$$

$$(3\sigma - 1)(n-1) + (2\sigma - 1)$$

$$n \log n - n + 1$$

Private Set Intersection Results



Some Other Results

	Problem	Best Previous Result	Our Result	Speedup
USENIX Security 2011	**Hamming Distance** (Face Recognition, Genetic Dating) – two 900-bit vectors	213s [SCiFI, 2010]	**0.051s**	**4176**
USENIX Security 2011	**Levenshtein Distance** (genome, text comparison) – two 200-character inputs	534s [Jha+, 2008]	**18.4s**	**29**
USENIX Security 2011	**Smith-Waterman** (genome alignment) – two 60-nucleotide sequences	[Not Implementable]	**447s**	-
USENIX Security 2011	**AES Encryption**	3.3s [Henecka, 2010]	**0.2s**	**16.5**
NDSS 2011	**Fingerprint Matching** (1024-entry database, 640x8bit vectors)	~83s [Barni, 2010]	**18s**	**4.6**

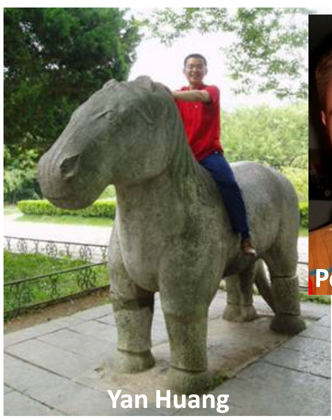
Scalable: 1 Billion gates evaluated at ~100,000 gates/second on laptop

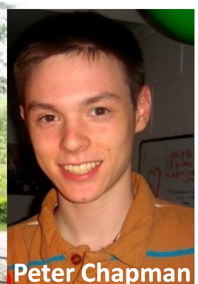
Demo!

Private Set Intersection on Android Devices

http://MightBeEvil.com/mobile/

Peter Chapman and Yan Huang


Yan Huang
(UVa Computer Science PhD Student)

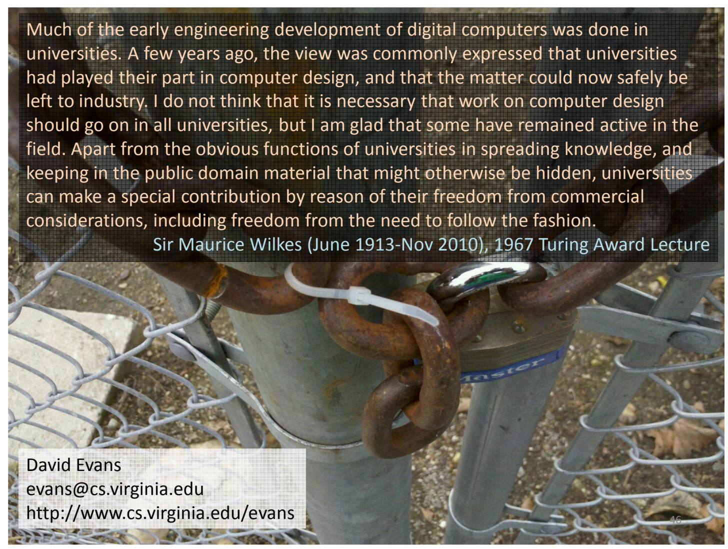

Peter Chapman
(UVa BACS 2012)


Jonathan Katz
(University of Maryland)


Aaron Mackey
(UVa Public Health Genomics)


Lior Makla
(UMd / Intel)

Funding: **NSF**, **MURI** (AFOSR)
Android toys: **Google**



Much of the early engineering development of digital computers was done in universities. A few years ago, the view was commonly expressed that universities had played their part in computer design, and that the matter could now safely be left to industry. I do not think that it is necessary that work on computer design should go on in all universities, but I am glad that some have remained active in the field. Apart from the obvious functions of universities in spreading knowledge, and keeping in the public domain material that might otherwise be hidden, universities can make a special contribution by reason of their freedom from commercial considerations, including freedom from the need to follow the fashion.

Sir Maurice Wilkes (June 1913-Nov 2010), 1967 Turing Award Lecture

David Evans
evans@cs.virginia.edu
http://www.cs.virginia.edu/evans

Introduction to
Computing
Explorations in Language, Logic, and Machines
Spring 2010

David Evans
University of Virginia

Shameless Plug

www.computingbook.org