Mobile Private Contact Discovery at Scale

Jan GOETTE, Mori Lab/Humboldt University of Berlin

June 2019 presentation on [kales1]

Contact Discovery

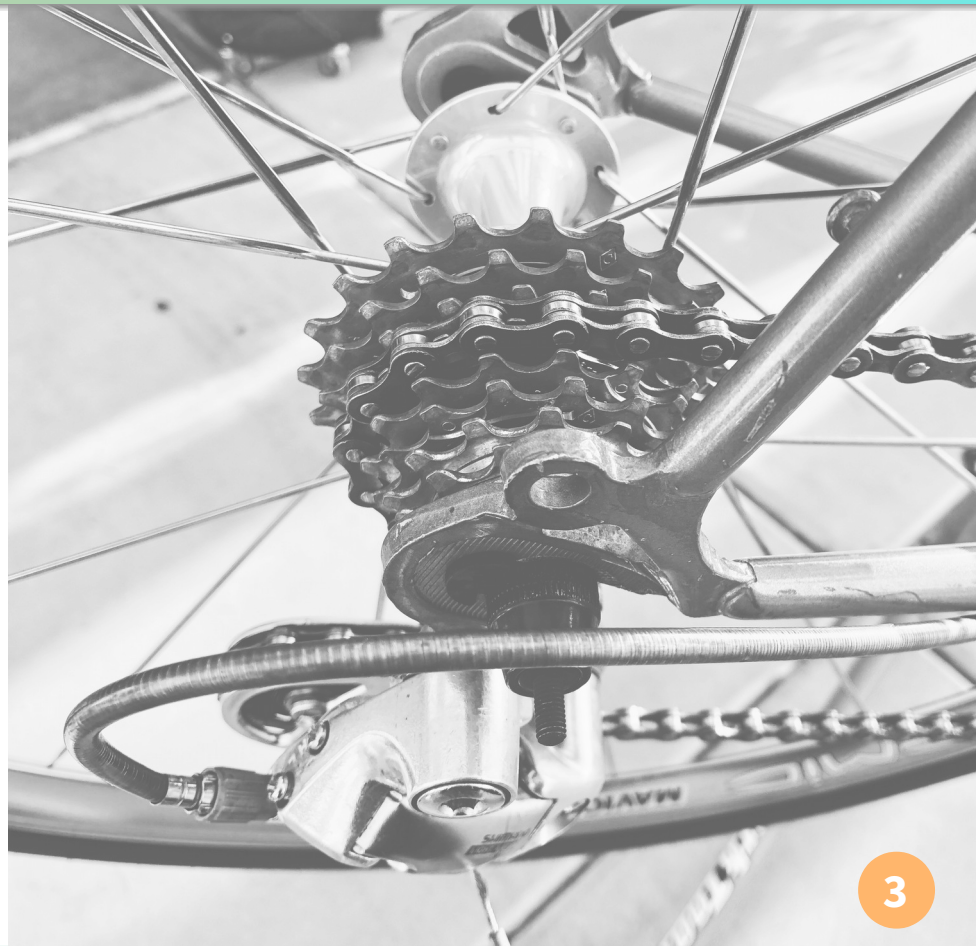
The Issue

- You're running a messenger (Line, Whatsapp, ...)
- Someone installs it
- How do you find out which of their contacts to show?

Contact Discovery

The State of the Art

- Upload all contacts to cloud
- Maybe sprinkle some hashes (!)
- Compare against database
- Download matches
- **Optional:** Use SGX (signal)



Private Set Intersection

Private Set Intersection

The Issue

- Two computers have two sets
- They want to compute their intersection
- **They must not learn anything about each other's sets other than that**
- Subfield of Secure Multi-Party Computation (SMPC)

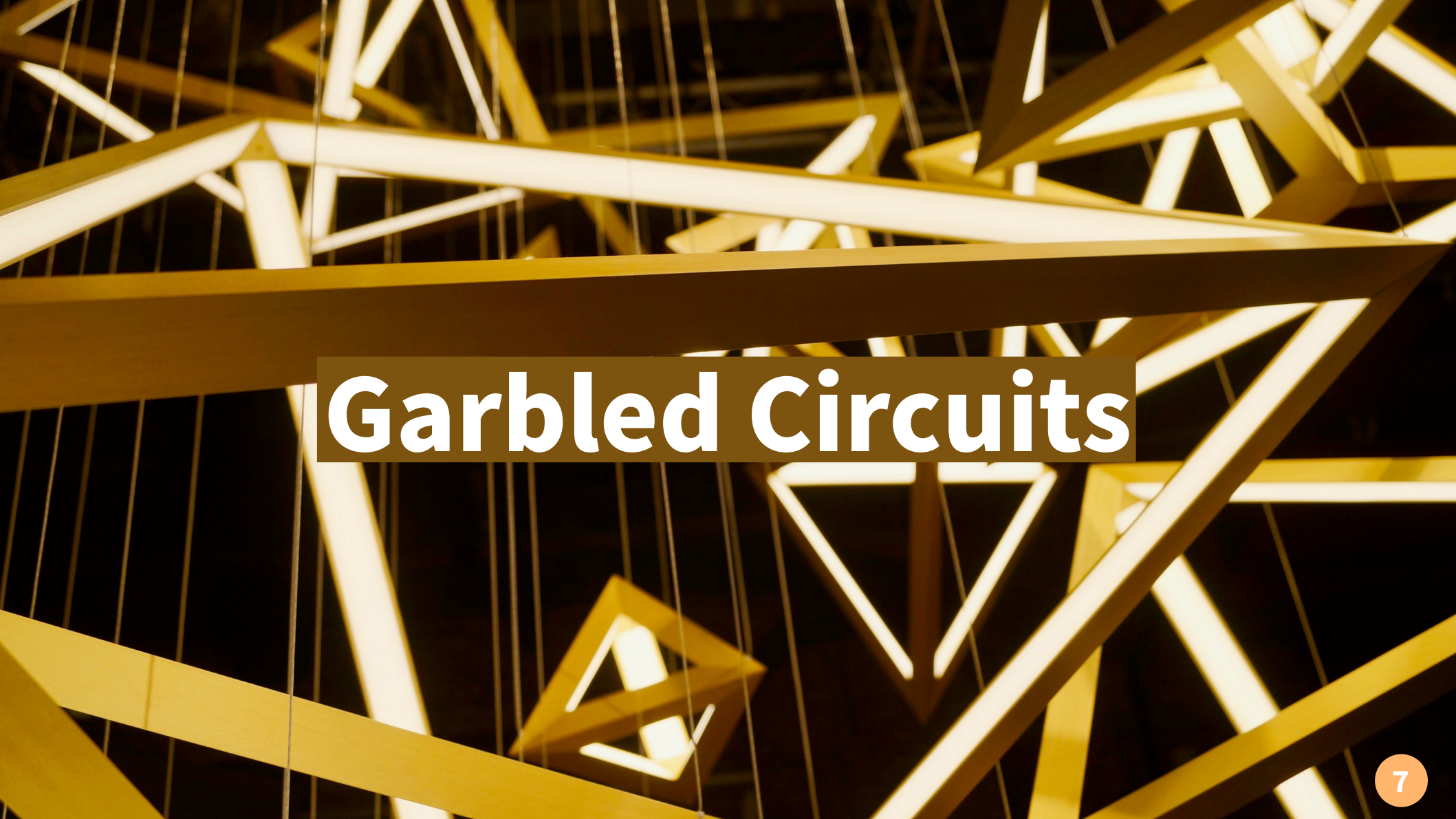
How do?

- Based on Oblivious Transfer (OT)
- Alice has X_0, X_1, Bob retrieves X_i without telling Alice i or learning X_{1-i}
- **Caution:** we have highly asymmetric set sizes [kiss1]

Private Set Intersection for Contact Discovery

One slight problem.

- Nobody is using this
- because it is too slow!
- [kales1] make it fast.


Garbled Circuits

Foundations: Garbled Circuits

Circuits

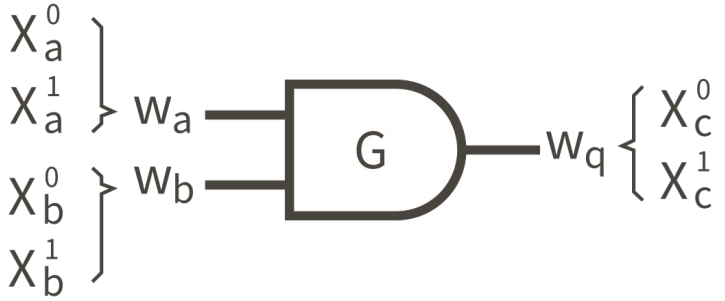
- “Circuit” here as in boolean circuit of gates such as AND, OR, NOT
- In practice: compiled to 2-input AND, XOR gates
- See Logic 101

The Algorithm

1) Alice compiles the circuit C
2) Alice garbles the circuit C→C'
3) Alice sends bob C' and her encrypted inputs
4) Bob has Alice encrypt his inputs using OT
5) Bob evaluates the circuit
6) Alice and Bob decrypt the output

Circuit garbling

- Encrypt output wire label for input (x, y) with input wire labels (X_a^x, X_b^y)
- Evaluator can only decrypt output label if they know corresponding input labels



b \ a	0	1
0	0	0
1	0	1

→

b \ a	X_a^0	X_a^1
X_b^0	X_c^0	X_c^0
X_b^1	X_c^0	X_c^1

→

b \ a	X_a^0	X_a^1
X_b^0	$E_{X_a^0, X_b^0}(X_c^0)$	$E_{X_a^1, X_b^0}(X_c^0)$
X_b^1	$E_{X_a^0, X_b^1}(X_c^0)$	$E_{X_a^1, X_b^1}(X_c^1)$

Foundations: Oblivious Pseudo-Random Functions (OPRFs)

OPRF What?

- Consider PRF $F_k(x)$, such as HMAC
- Alice chooses k, Bob chooses x
- Bob computes $F_k(x)$ without Bob learning k or Alice learning x

OPRF How?

- Gajillions of variants, we only consider one.
- Alice keys cipher (e.g. AES) and compiles it to Garbled Circuit
- Bob evalued Garbled Circuit on x


Private
Contact
Discovery

Assembling an Private Contact Discovery Algorithm

Simple PCD

- Server encrypts its database locally using same PRF as in OPRF
- Server sends encrypted DB to client
- Client encrypts its contacts using OPRF
- Client matches output against DB

Shortcoming

- Large database → large resource usage

Cuckoo Filters

Probabilistic data structure similar to Bloom Filters

- Allows insert, delete, lookup
- Lookup returns “**maybe**” or “**definitely not**”
- Trade-off **Space—False positive rate**

Optimizing the Private Contact Discovery Algorithm

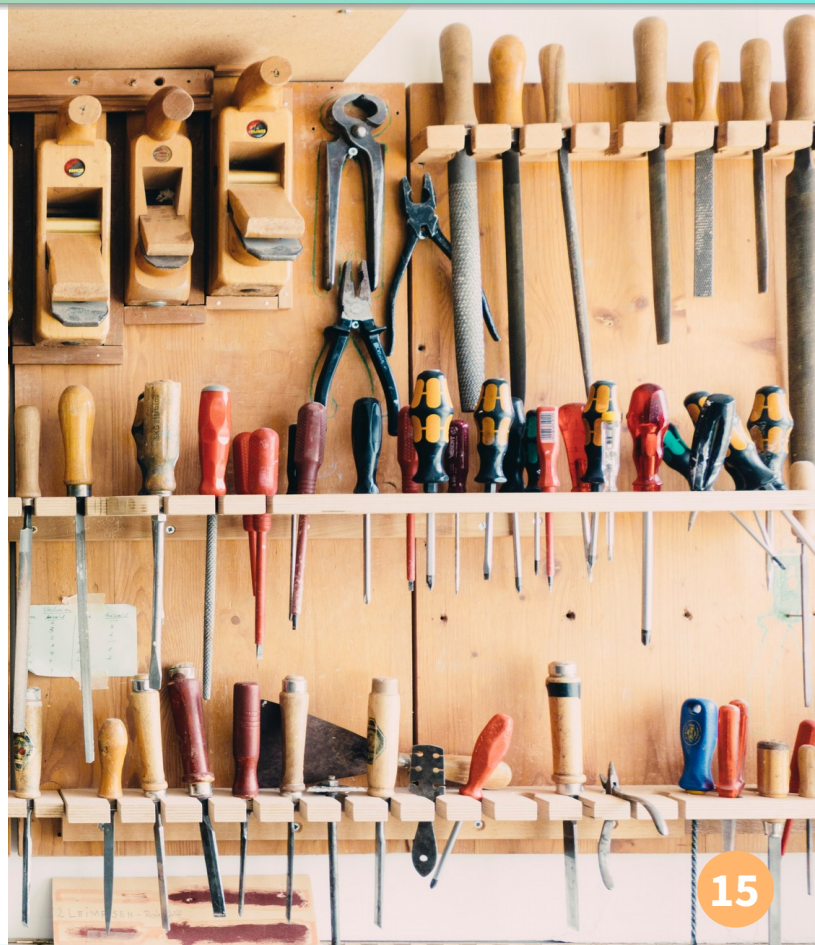
Use cuckoo filter for DB transfer!

- Uses asymptotically small amount of storage per server contact
- Fast lookup
- Allow delta-updates

Key Improvements of [kales1] for a Practical System

- Cuckoo Filter instead of Bloom Filter
- **Fine-tune Cuckoo Filter** parameters
- PSI using cipher-based OPRF
- Replace AES with lightweight cipher **LowMC**
- Compress cuckoo filters the smart way
- Use delta-updates for cuckoo filters
- **Limitation:** Semi-honest adversary model!

Result: Check 1024 contacts against 2^{28} in db on a smartphone over WiFi in 3s



Take-aways?

- **Small improvements can have large cumulative impact**
  - See: Any large software project, e.g. Firefox
- **Improving academic experiments is worthwhile**
  - you might end up with something practical
- **Don't handicap your designs, choose parameters with care**
  - Selecting a good set of parameters might not take much time, and is important for the high-level perception of your work by others.
- **Now, in 2019, there's no excuse anymore to do privacy-invasive contact discovery**
  - Seriously!


YOU'LL GET IT
EVENTUALLY
marmaladelondon.co.uk

Questions

Research Ideas!

Research directions

- **Exploiting geographic correlation** of contact graph neighborhood
  - Have several PSI servers serving large geographic areas (e.g. “east-asia”, “Japan”)
  - Perform PSI contact discovery starting from closest server
- **Improve UI through incremental refinement**
  - Split 32-bit cuckoo filter data into 33 datasets: (0) The empty bucket bitmap and (1-32) the bitmaps of fingerprint bits 0-31
  - Download the empty bucket bitmap and bitmaps 0-8 before starting PSI. Construct an 8-bit cuckoo filter, and perform local matching with high false-positive rate.
  - After PSI, download the remaining bitmaps, and update the UI with updated matches from bit-by-bit improved cuckoo filter. I.e. start with high error rate that decreases (contact list being updated) while the rest of the cuckoo filter is downloaded.

Oblivious Transfer (see [chou1])

- **Alice has X_0, X_1, Bob retrieves X_i without telling Alice i or learning X_{1-i}**
- Here: Based on DH assumption
- Asymmetric crypto, so comparatively slow
- Can be pre-computed, with actual payloads masked later using XOR

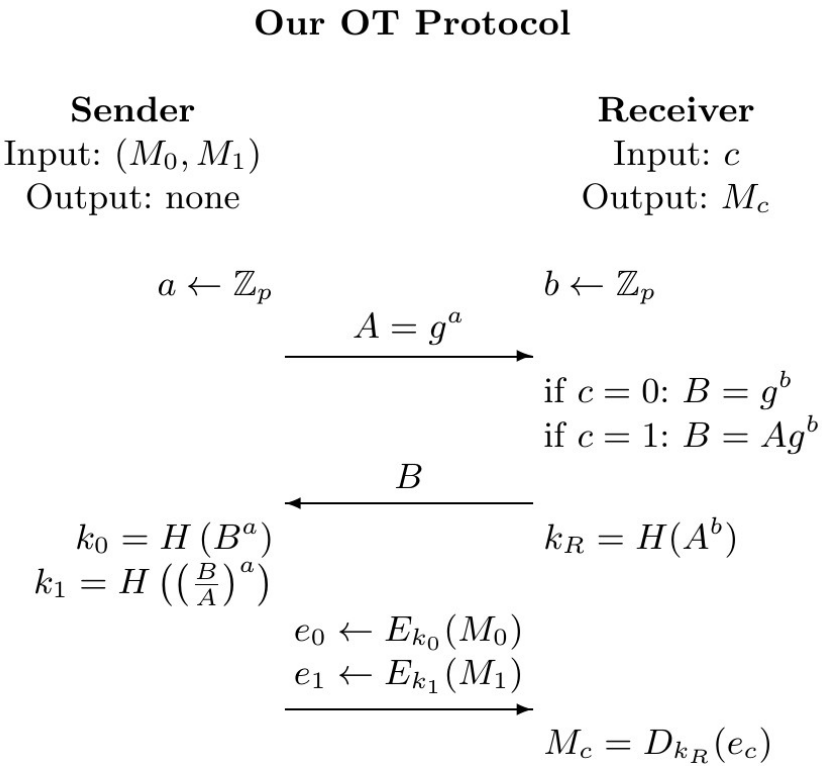


Figure 1. Our protocol in a nutshell

Image sources

- Title slide photo by Toa Heftiba on Unsplash
- Speech bubble photo by Jason Leung on Unsplash
- Bicycle wheel photo by rick on Unsplash
- Telephones photo by Pavan Trikutam on Unsplash
- Eggs in carton photo by Miguel Andrade on Unsplash
- Neon sign photo by Nigel Tadyanehondo on Unsplash
- Tetrahedron lamp photo by James Orr on Unsplash
- Library photo by Susan Yin on Unsplash
- Workshop tools photo by Barn Images on Unsplash
- Foot stand photo by Barn Images on Unsplash
- Sad teddy photo by Trym Nilsen on Unsplash
- Greenhouse photo by Trym Nilsen on Unsplash
- Kitten photo by Tran Mau Tri Tam on Unsplash

References

- [chou1] Tung Chou and Claudio Orlandi. The Simplest Protocol for Oblivious Transfer. In LATINCRYPT, volume 9230 of LNCS, pages 4058. Springer, 2015.
- [kiss1] Ágnes Kiss, Jian Liu, Thomas Schneider, N. Asokan, and Benny Pinkas. Private Set Intersection for Unequal Set Sizes with Mobile Applications. PoPETs, 2017(4):177197, 2017.
- [pinkas1] Benny Pinkas, Thomas Schneider, Nigel P. Smart, and Stephen C. Williams. Secure Two-Party Computation Is Practical. In ASIACRYPT, volume 5912 of LNCS, pages 250267. Springer, 2009.
- [kales1] Daniel Kales , Christian Rechberger, Thomas Schneider, Matthias Senker and Christian Weinert. Mobile Private Contact Discovery at Scale. In Cryptology ePrint Archive, Report 2019/517. 2019.