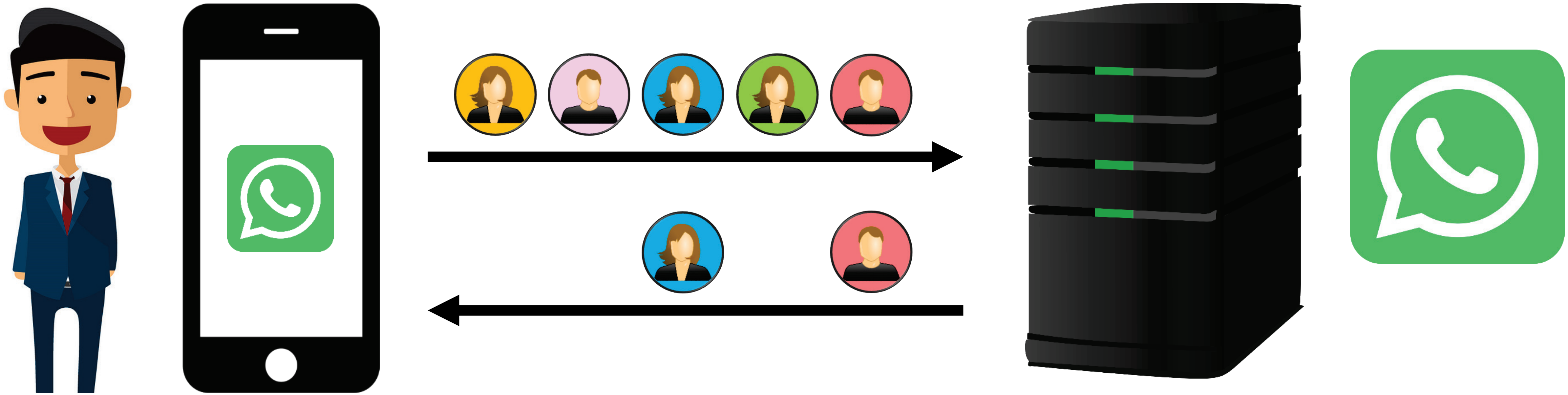
Mobile Private Contact Discovery at Scale

Daniel Kales (TU Graz), Christian Rechberger (TU Graz), Thomas Schneider (TU Darmstadt), Matthias Senker (TU Darmstadt), Christian Weinert (TU Darmstadt)



https://contact-discovery.github.io/

Agenda

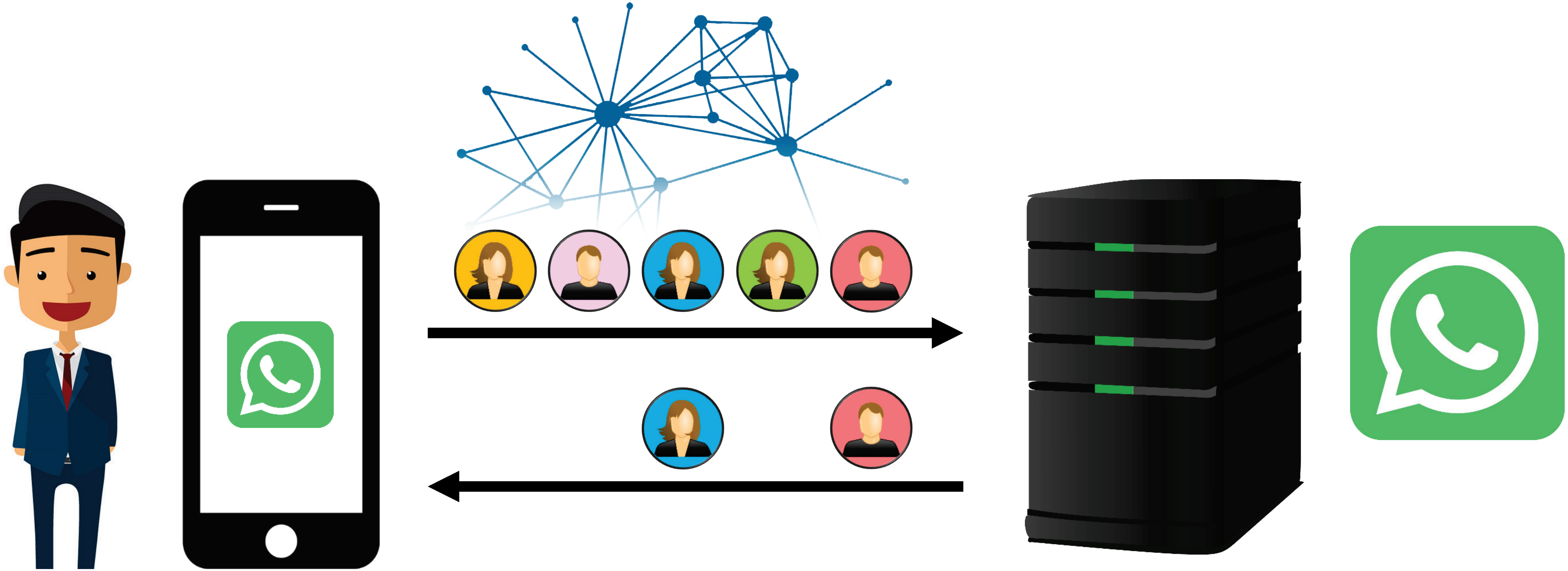
TECHNISCHE UNIVERSITÄT DARMSTADT

1. Mobile contact discovery in practice
2. Improved *unbalanced* **Private Set Intersection (PSI)** protocols
3. Native MPC protocol implementations on *mobile* platforms

https://contact-discovery.github.io/

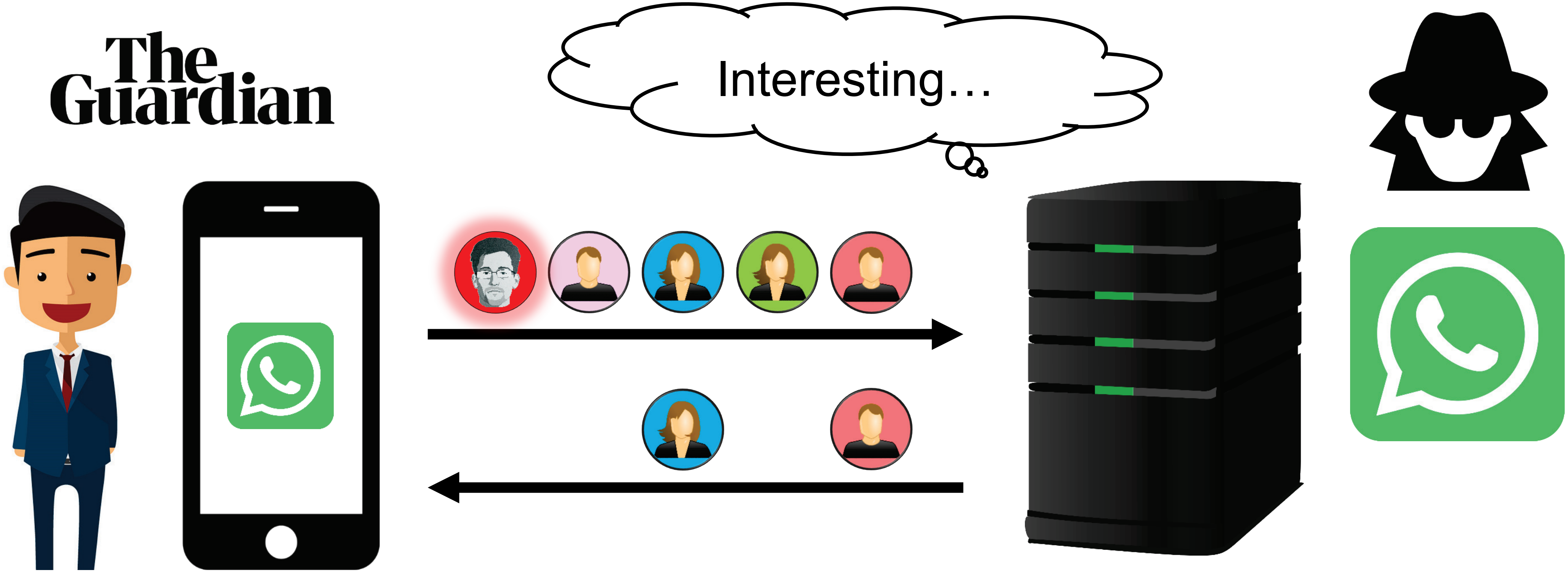
ENCRYPTO CRYPTOGRAPHY AND PRIVACY ENGINEERING

Mobile Contact Discovery – Privacy Concerns?



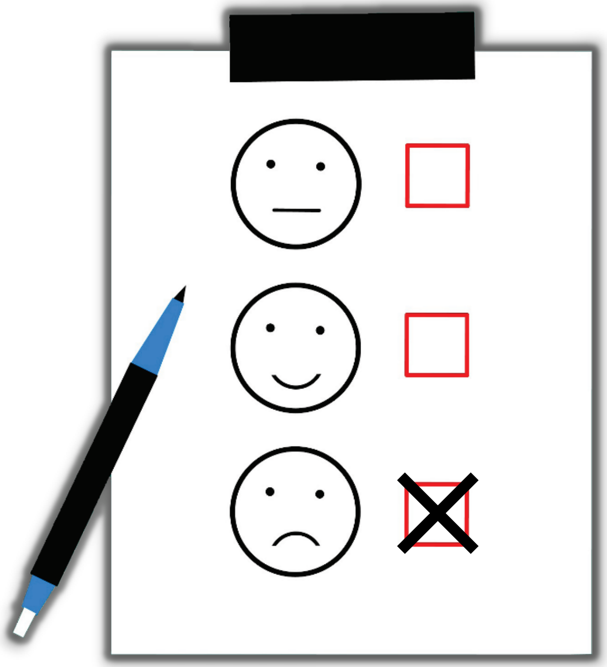
https://contact-discovery.github.io/

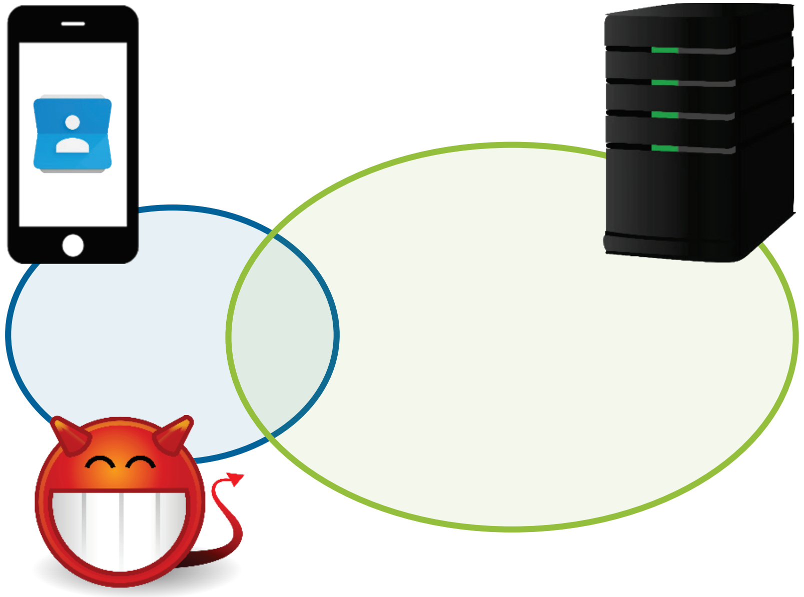
Mobile Contact Discovery – Privacy Concerns!



https://contact-discovery.github.io/

Our Contributions





Survey of secure mobile messaging applications

Optimized OPRF-based **unbalanced PSI** protocols

Native implementations in C/C++ utilizing **ARMv8-A** instruction sets

No proper privacy protection during contact discovery

Cuckoo filter **compression**, new OT precomputation, **LowMC** for OPRF

1000x faster GC evaluation, Signal integration, ~5s online phase

Contact Discovery in “Secure” Mobile Messaging Applications

SURVEY

Contact Discovery in “Secure” Mobile Messaging Applications

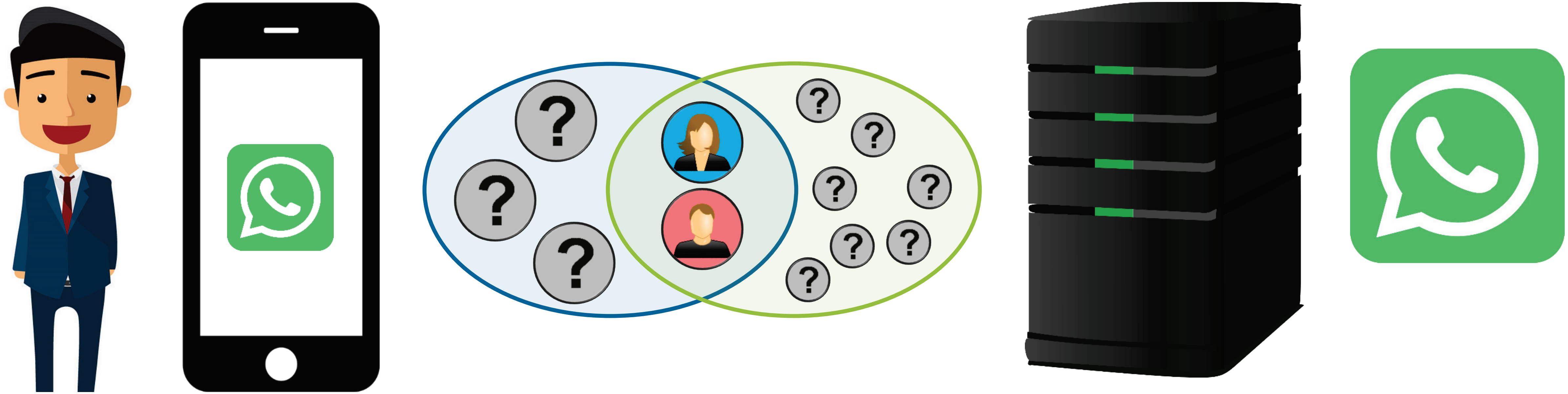
Messenger	Naïve Hashing	Analysis Technique
Confide*	✔	Privacy Policy
Dust*	✗	Network Traffic
Eleet*	✗	Privacy Policy
G DATA Secure Chat	✔	Network Traffic
Signal (legacy / non-SGX)	✔	Source Code
SIMSme	✔	Network Traffic
Telegram	✗	Privacy Policy
Threema	✔	Privacy Policy
Viber	✗	Privacy Policy
WhatsApp	✗	Privacy Policy
Wickr Me	✔	Privacy Policy
Wire	✔	Privacy Policy

* contact discovery is optional

Unbalanced PSI Protocols – Related Work – Precomputation Form

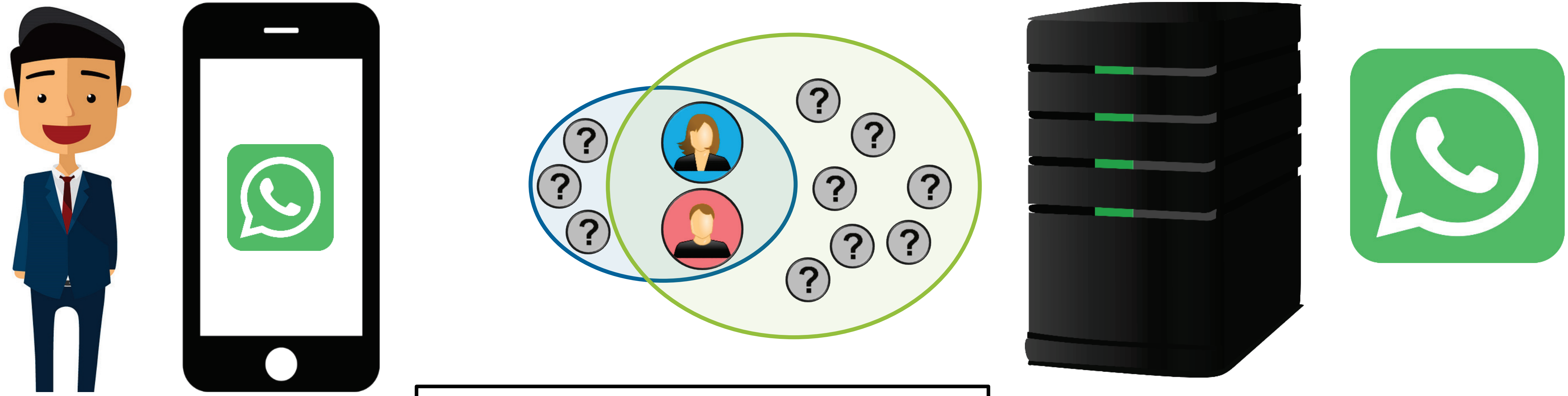
PSI FOR MOBILE CONTACT DISCOVERY

Private Set Intersection (PSI) Protocols



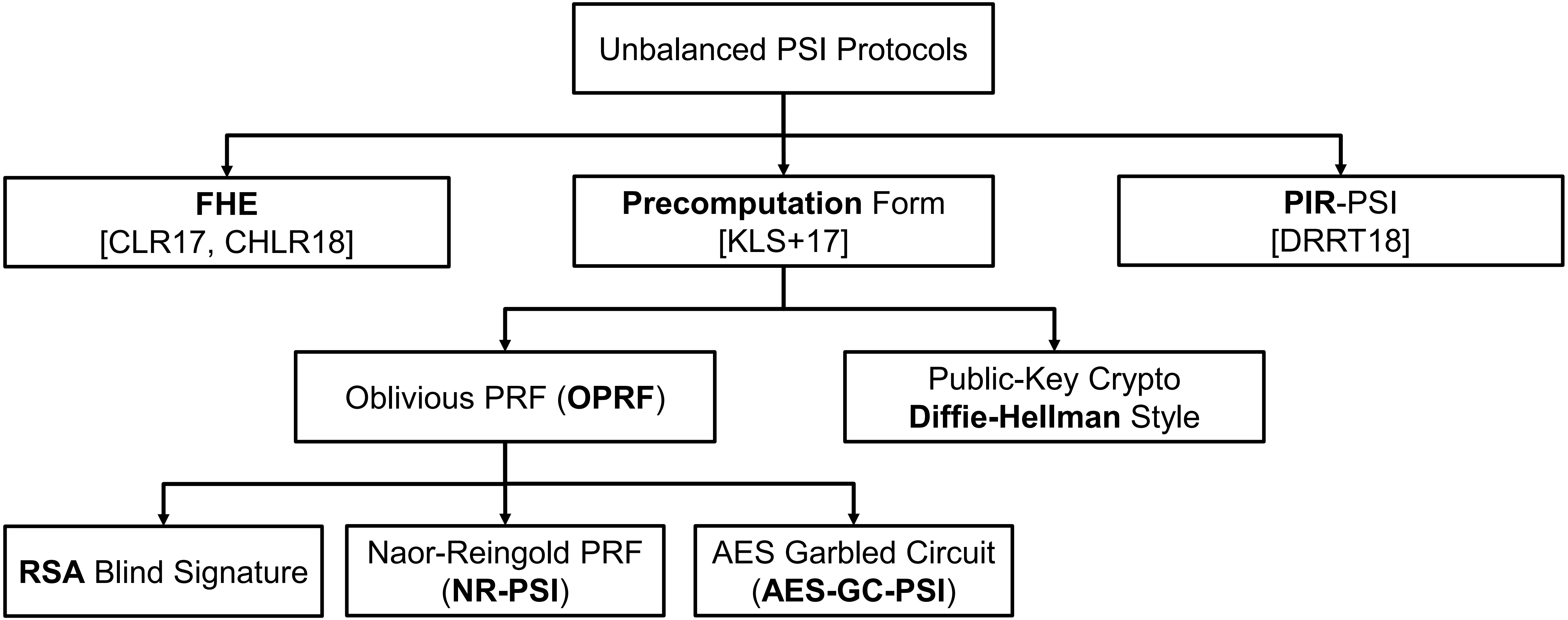
Communication complexity:
$O(|\text{Client}| + |\text{Server}|)$ in **online** phase!

Unbalanced PSI Protocols



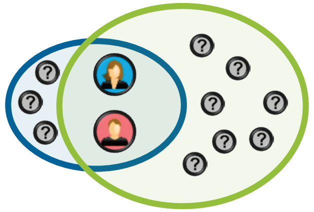
Communication complexity:
$O(|\text{Client}|)$ in **online** phase
$O(|\text{Server}|)$ in **setup** phase

Related Work



OPRF-Based Unbalanced PSI Protocols in Precomputation Form

Contacts $c_{i\in\{1,\ldots,n\}}$



1. Base Phase $O(|\text{Client}|)$

OT Precomputation

Generate secret key k

(Build Garbled Circuits GC_i)

$\left(\xleftarrow{GC_i} \right)$

2. Setup Phase $O(|\text{Server}|)$

Encrypt all contacts with key k and insert them into Cuckoo filter CF

$\xleftarrow{CF}$

Store Cuckoo filter CF

3. Online Phase $O(|\text{Client}|)$

Run OPRF for all contacts c_i

$c_i \rightarrow$ OPRF $\leftarrow k$

$e_i \leftarrow$ OPRF

Check if e_i is in CF

Cuckoo Filter Compression – LowMC

PROTOCOL OPTIMIZATIONS

Cuckoo Filter Compression

Cuckoo Filter

0	a		
1		b	
2			
3	c	d	
4		e	f
5	g		
6	h	i	
7	j		

Bitmap

1	0	0
0	1	0
0	0	0
1	1	0
0	1	1
1	0	0
1	1	0
1	0	0

Tag List

(a, b, c, d, e, f, g, h, i, j)

Compression ratio ≈ Load factor

- Realistic parameters: tag size 32 / 42 bit and bucket size 3 for FPP 2^{-30} / 2^{-40}
- More efficient updates: 4.3x less communication

More Efficient PRF for GC-PSI

TECHNISCHE UNIVERSITÄT DARMSTADT

- Free XOR [KS08] optimization for Yao's GC protocol allows "free" evaluation of XOR gates
  - ➢ Use LowMC [ARS+15] instead of AES
    - Highly parametrizable block cipher for MPC and FHE applications

PRF	Block Size	Key Size	#S-Boxes	Data Complexity	#Rounds	#ANDs
LowMC	128	128	42	2^{64}	13	1638
	128	128	31	2^{64}	13	1209
	128	**128**	**1**	$\mathbf{2^{64}}$	**208**	**624**
	128	128	1	2^{32}	192	576
	128	128	1	2^{128}	287	861
AES-128	128	128	16	2^{128}	10	5120

8.2x Comm. Improvement

ENCRYPTO CRYPTOGRAPHY AND PRIVACY ENGINEERING

TECHNISCHE UNIVERSITÄT DARMSTADT

Open Source!

https://contact-discovery.github.io/

ARMv8-A Instruction Sets

IMPLEMENTATION

ENCRYPTO CRYPTOGRAPHY AND PRIVACY ENGINEERING

ARMv8-A Instruction Sets for Native Implementations

TECHNISCHE UNIVERSITÄT DARMSTADT

- Malicious Secure OT Extension Protocols: libOTe [Rin]
  - Heavily optimized for x86 architecture
  - ➢ Ported to ARMv8-A while maintaining compatibility with x86 counterpart

- Yao's GC protocol [Yao86] with **fixed-key AES garbling** [BHKR13]
  - ARMv8 Cryptography Extensions (CE) provide hardware instructions for **AES**, SHA-1, and SHA-2
    - ➢ **35x faster AES evaluations** compared to standard software implementation
  - ARMv8 NEON instruction set for vector operations on 128-bit registers
    - ➢ Efficiently work with 128-bit wire labels

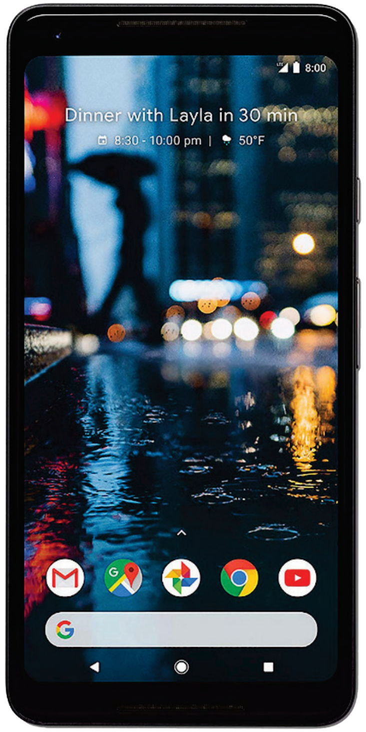
arm NEON

➢ **1000x faster GC evaluation** than Java implementation of [KLS+17] based on ObliVM [LWN+15]

Setup – Benchmarks – Protocol Extensions

EVALUATION

Setup – WiFi

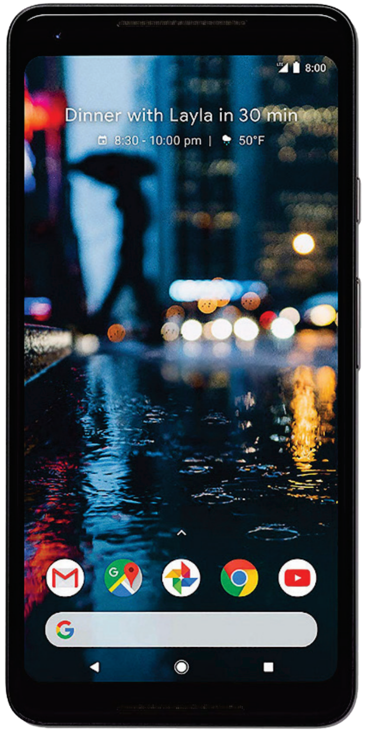


Google Pixel 2 XL
Snapdragon 835 CPU @ 2.45GHz
4GiB of RAM

IEEE 802.11ac WiFi
230Mbit/s down-/upload
70ms RTT

Commodity Laptop
Intel Core i7-4600U CPU @ 2.6GHz
16GiB of RAM

Setup – LTE



Google Pixel 2 XL
Snapdragon 835 CPU @ 2.45GHz
4GiB of RAM

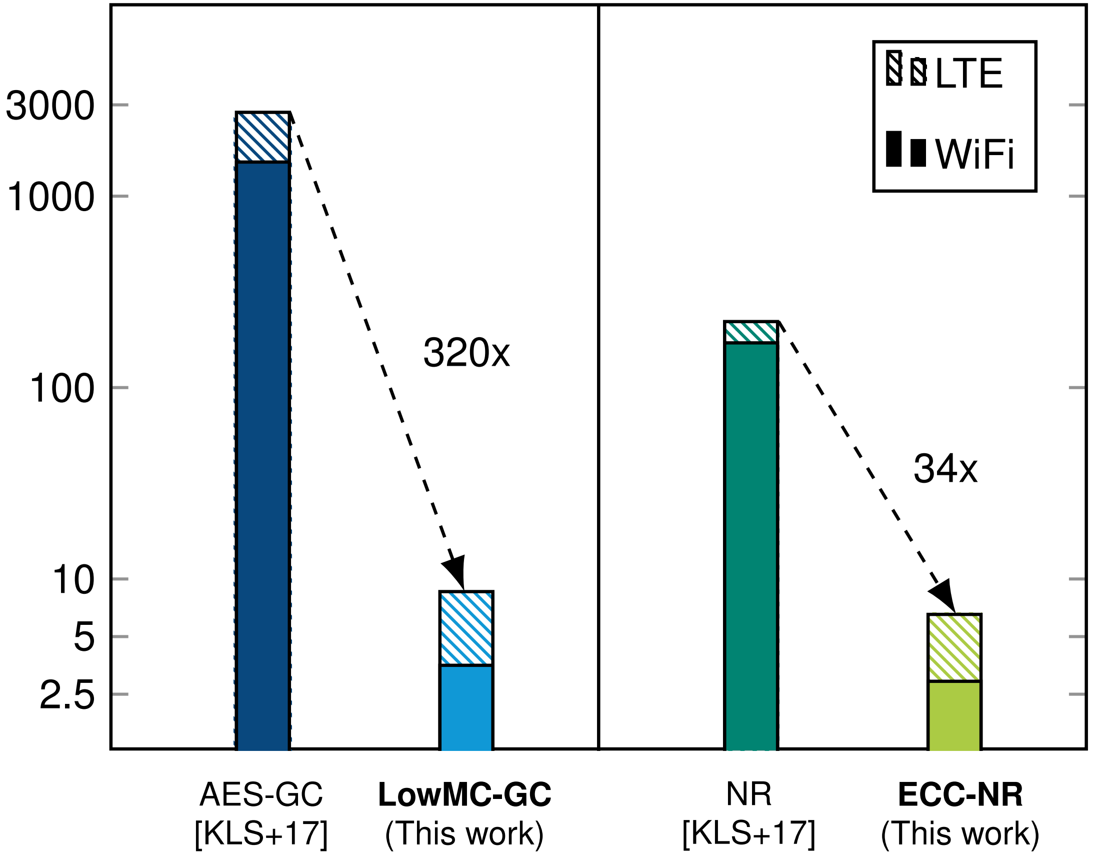
LTE™

42Mbit/s download
4Mbit/s upload
80ms RTT

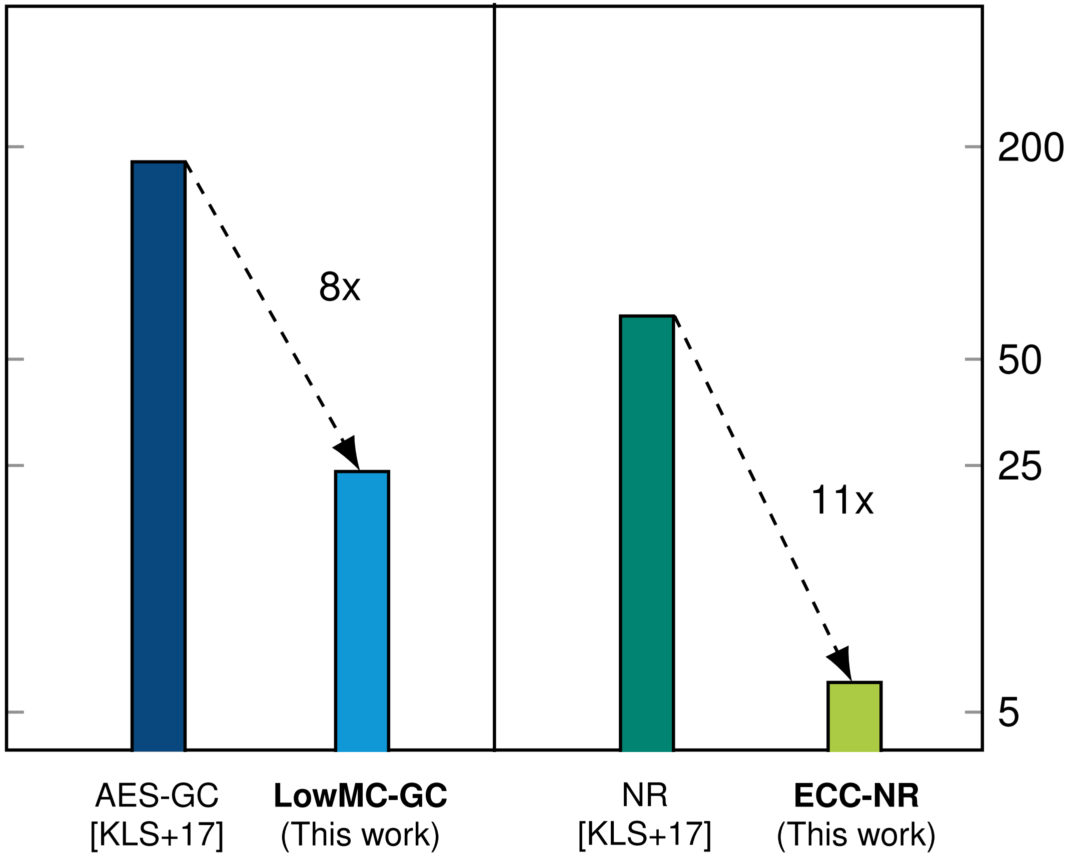
Commodity Laptop
Intel Core i7-4600U CPU @ 2.6GHz
16GiB of RAM

Benchmarks – Base + Online Phase (Checking 1k Client Contacts)

Run-Time (in seconds)

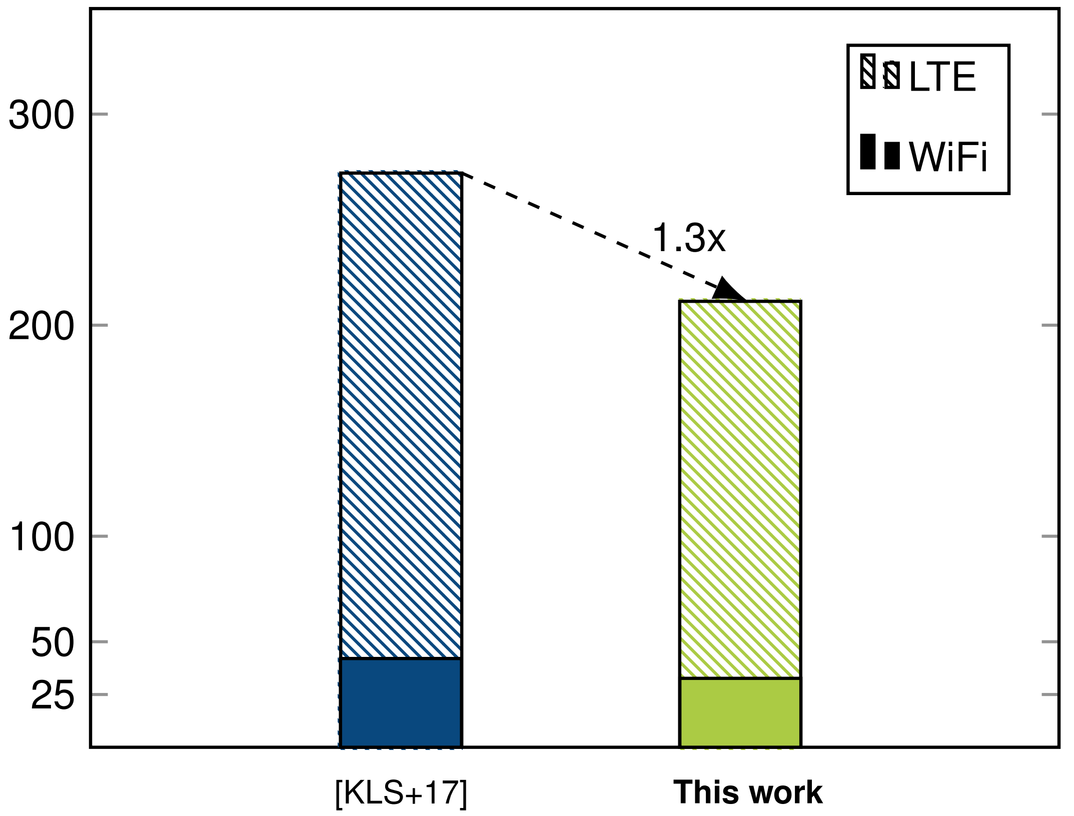


Communication (in MiB)

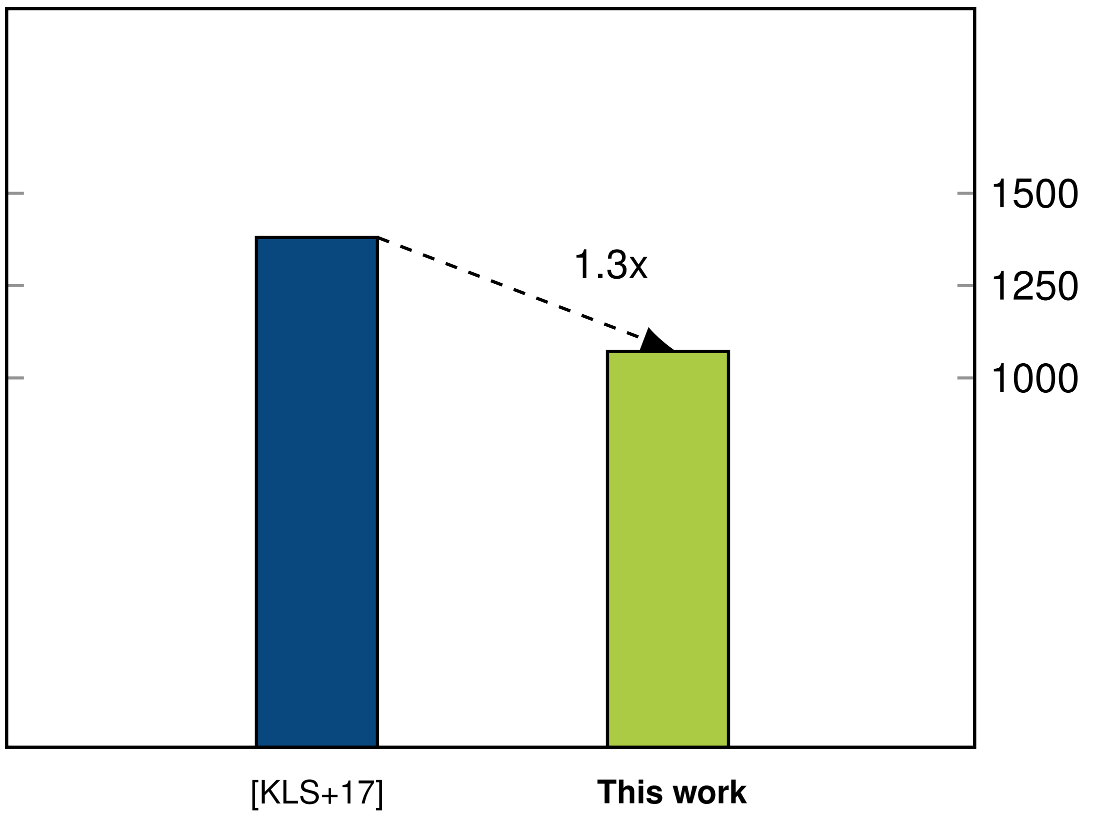


Benchmarks – Setup Phase (for $2^{28} = 268$M Server Contacts)

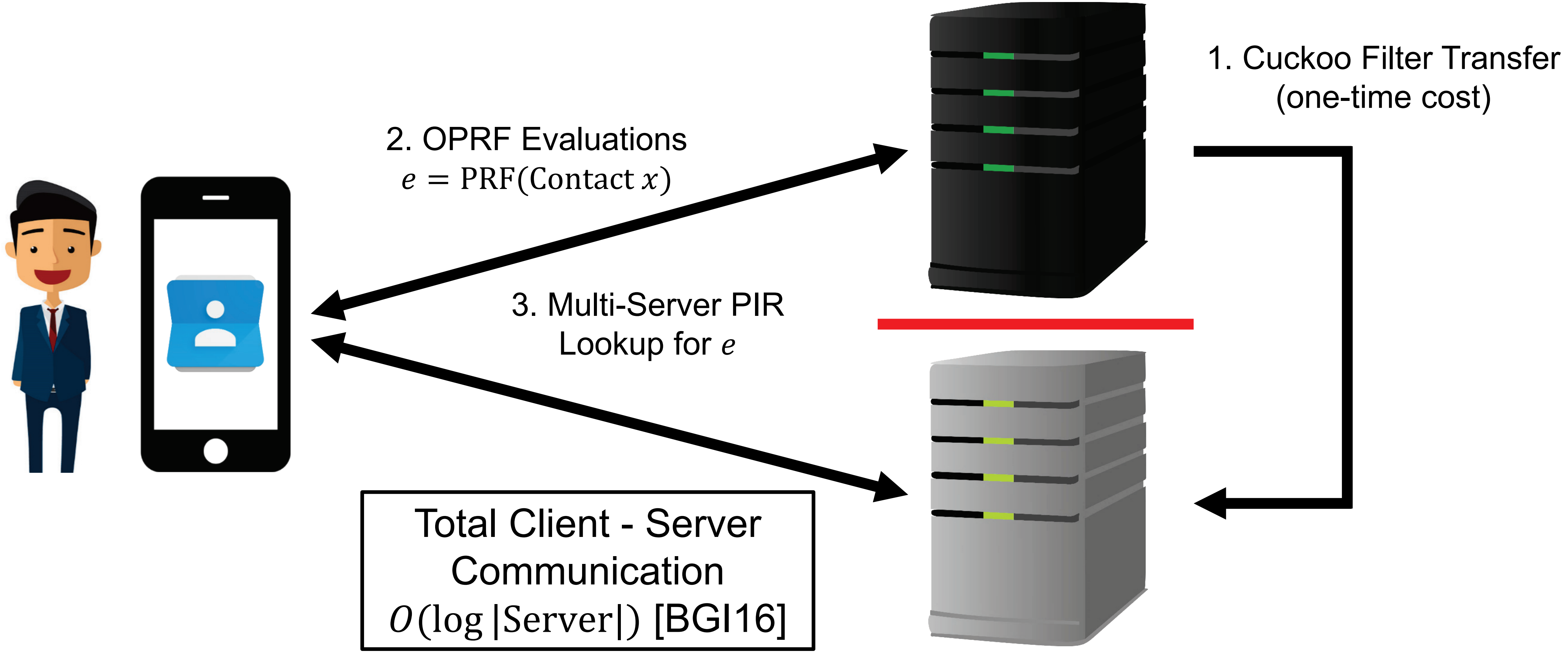
Run-Time (in seconds)



Communication (in MiB)



Protocol Extensions – Combination with Multi-Server PIR



CONCLUSION

Conclusion

- Most practical protocols and implementations for mobile **private** contact discovery at scale
- **General purpose** unbalanced PSI protocols
  - ➢ Mobile malware detection service, discovery of leaked passwords, etc.
- **Native Yao's GC implementation** on ARMv8-A

	Requirement	**This Work**
# Registered Users	> 1B	~ 250M
# Entries per Address Book	10k	1k
Latency	< 2s	> 30s (~ 5s online)
Communication	**< 10MiB**	> 1GiB (~ 6MiB online)

https://contact-discovery.github.io/

Conclusion

Thank You!

https://contact-discovery.github.io/

Bibliography (1/3)

TECHNISCHE UNIVERSITÄT DARMSTADT

[ALSZ17] Gilad Asharov, Yehuda Lindell, Thomas Schneider, and Michael Zohner. **More Efficient Oblivious Transfer Extensions**. In *JoC*. Springer, 2017.

[ARS+15] Martin R. Albrecht, Christian Rechberger, Thomas Schneider, Tyge Tiessen, and Michael Zohner. **Ciphers for MPC and FHE**. In *EUROCRYPT*. Springer, 2015.

[Bea95] Donald Beaver. **Precomputing Oblivious Transfer**. In *CRYPTO*. Springer, 1995.

[BGI16] Elette Boyle, Niv Gilboa, and Yuval Ishai. **Function Secret Sharing: Improvements and Extensions**. In *CCS*. ACM, 2016.

[BHKR13] Mihir Bellare, Viet Tung Hoang, Sriram Keelveedhi, and Phillip Rogaway. **Efficient Garbling from a Fixed-Key Blockcipher**. In *IEEE S&P*. IEEE Computer Society, 2013.

ENCRYPTO CRYPTOGRAPHY AND PRIVACY ENGINEERING

Bibliography (2/3)

[CHLR18] Hao Chen, Zhicong Huang, Kim Laine, and Peter Rindal. **Labeled PSI from Fully Homomorphic Encryption with Malicious Security**. In *CCS*. ACM, 2018.

[CLR17] Hao Chen, Kim Laine, and Peter Rindal. **Fast Private Set Intersection from Homomorphic Encryption**. In *CCS*. ACM, 2017.

[DRRT18] Daniel Demmler, Peter Rindal, Mike Rosulek, and Ni Trieu. **PIR-PSI: Scaling Private Contact Discovery**. In *PoPETs*. De Gruyter Open, 2018.

[KLS+17] Ágnes Kiss, Jian Liu, Thomas Schneider, N. Asokan, and Benny Pinkas. **Private Set Intersection for Unequal Set Sizes with Mobile Applications**. In *PoPETs*. De Gruyter Open, 2017.

[KS08] Vladimir Kolesnikov and Thomas Schneider. **Improved Garbled Circuit: Free XOR Gates and Applications**. In *ICALP*. Springer, 2008.

Bibliography (3/3)

[LWN+15] Chang Liu, Xiao Shaun Wang, Kartik Nayak, Yan Huang, and Elaine Shi. **ObliVM: A Programming Framework for Secure Computation**. In *IEEE S&P*. IEEE Computer Society, 2015.

[RdFA18] Amanda Cristina Davi Resende and Diego de Freitas Aranha. **Faster Unbalanced Private Set Intersection**. In *FC*. Springer, 2018.

[Rin] Peter Rindal. **libOTe: A fast, portable, and easy to use Oblivious Transfer Library**. https://github.com/osu-crypto/libOTe.

[Yao86] Andrew Chi-Chih Yao. **How to Generate and Exchange Secrets (Extended Abstract)**. In *FOCS*. IEEE, 1986.