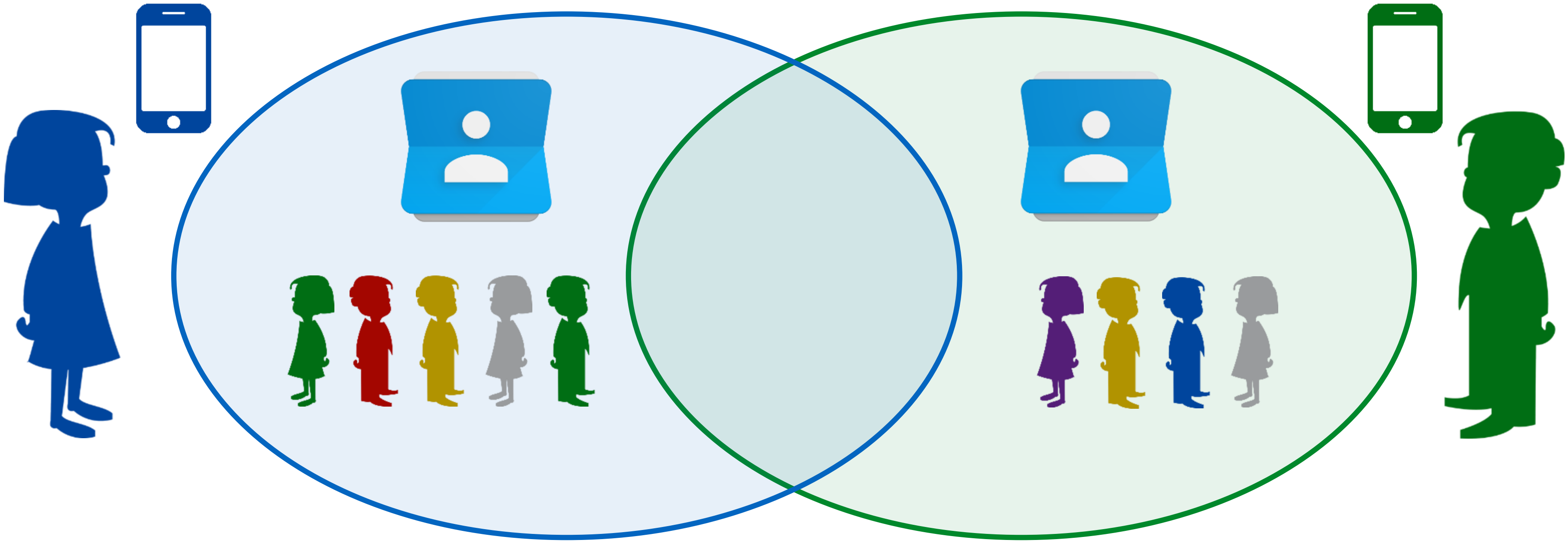
Private Set Intersection for Unequal Set Sizes with Mobile Applications

Ágnes Kiss (TU Darmstadt)

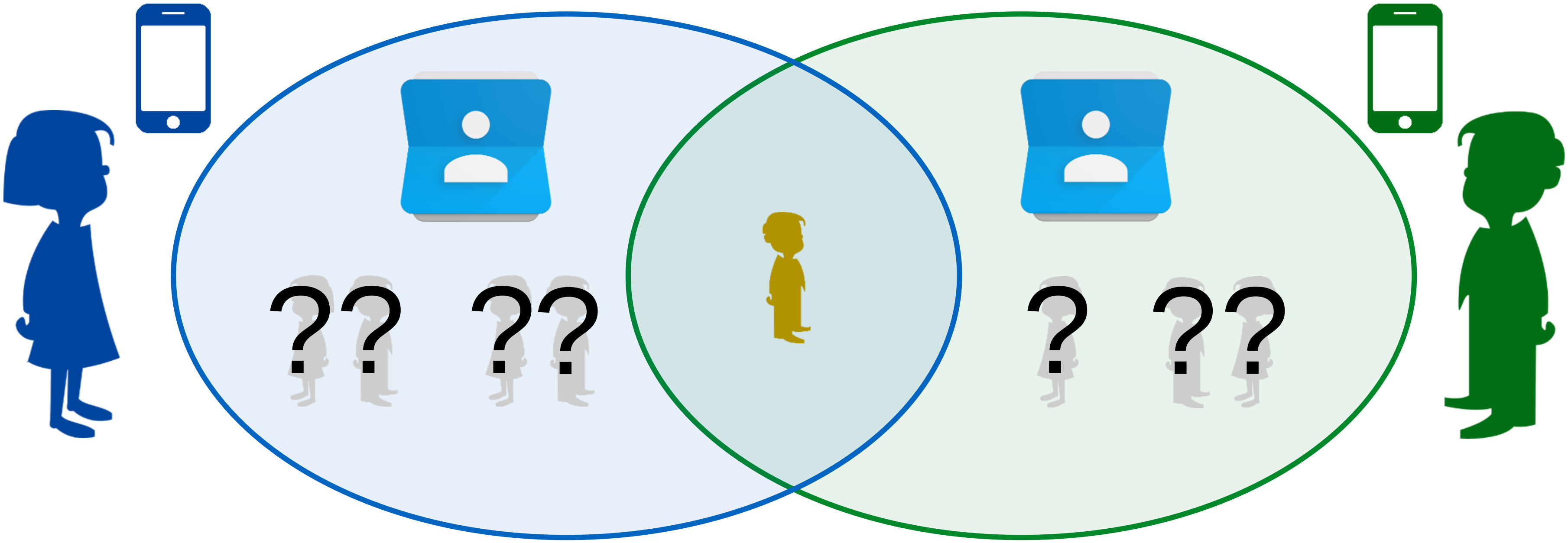
Jian Liu (Aalto University)
Thomas Schneider (TU Darmstadt)
N. Asokan (Aalto University)
Benny Pinkas (Bar-Ilan University)

Private Set Intersection (PSI)



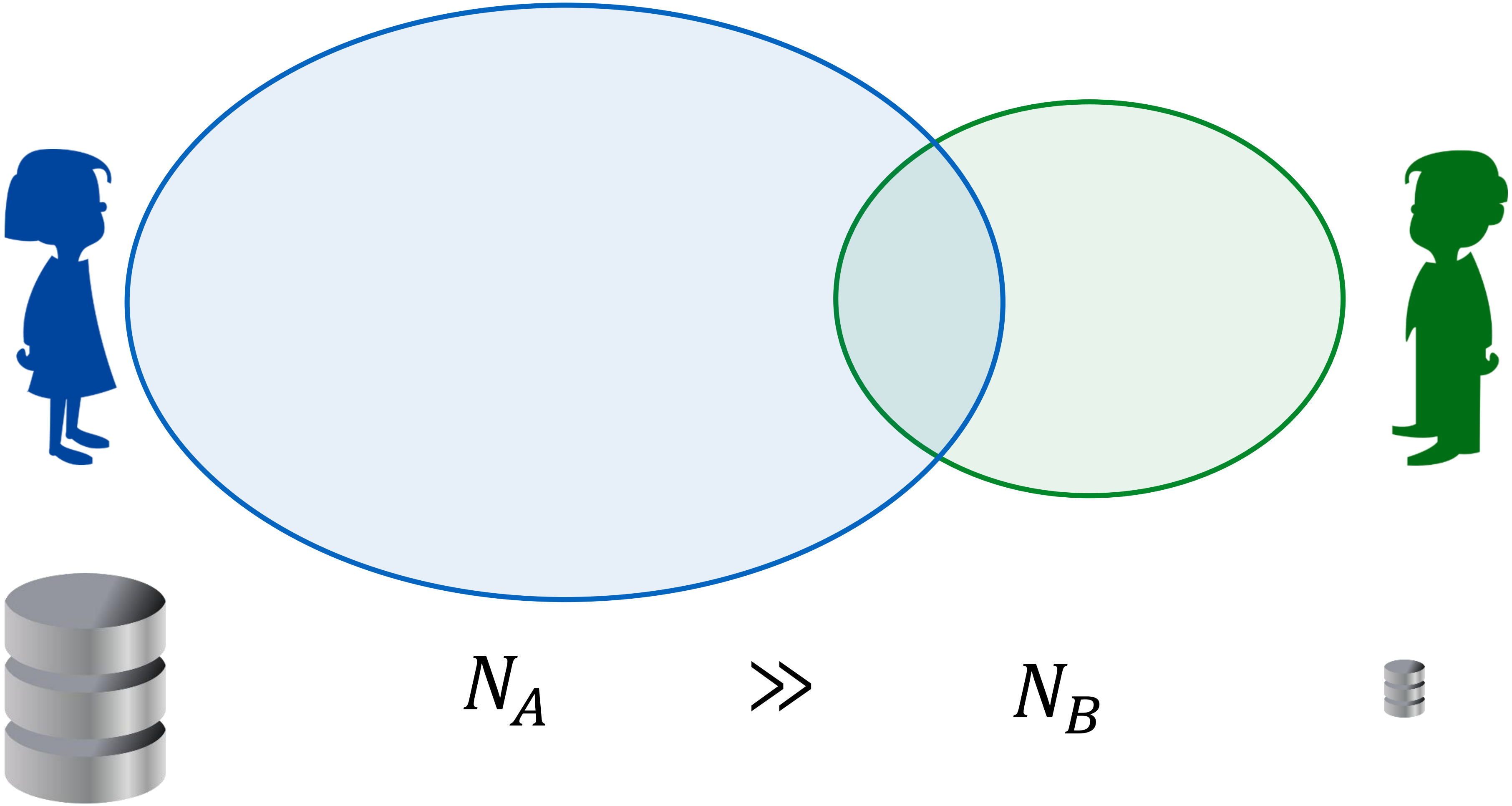
$$N_A \approx N_B$$

Private Set Intersection (PSI)

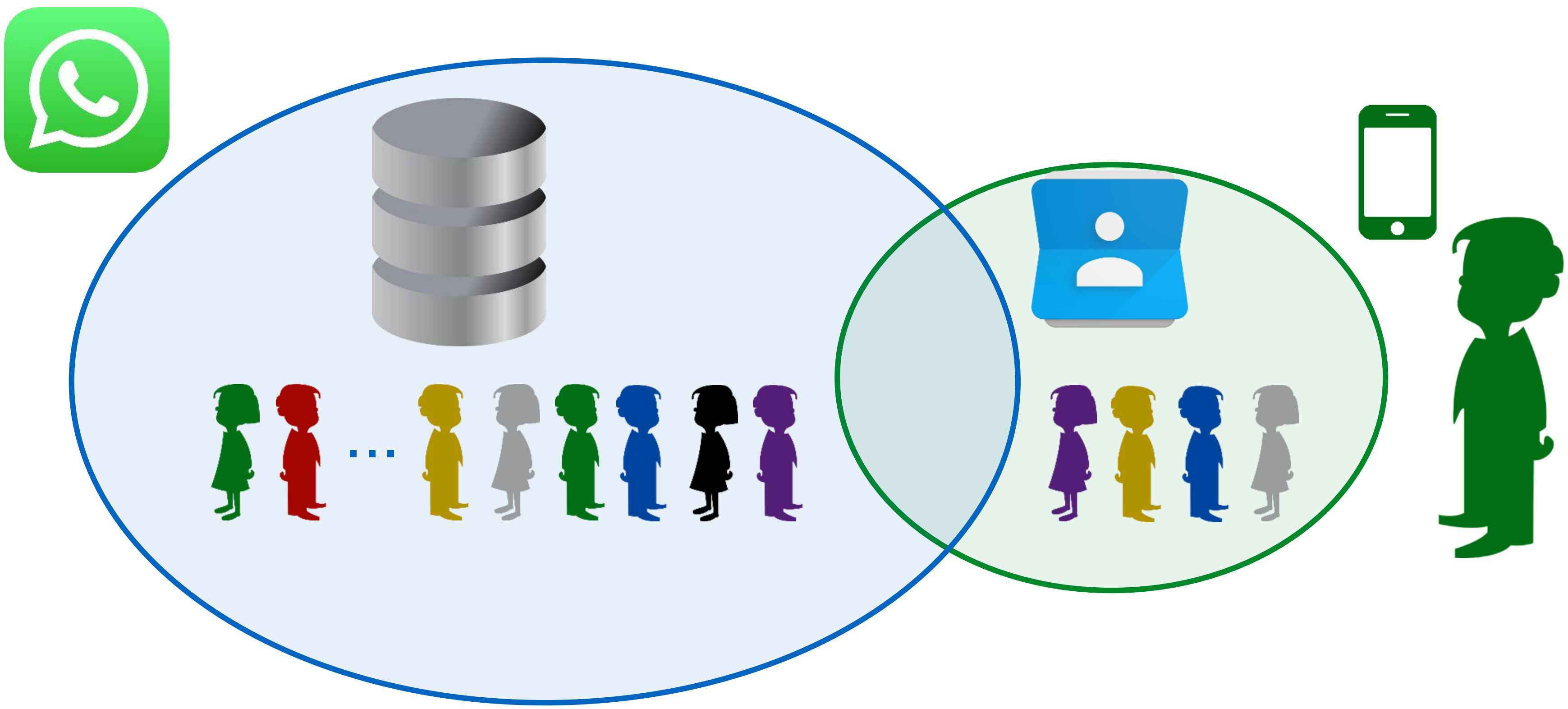


$$N_A \approx N_B$$

PSI with Unequal Set Sizes

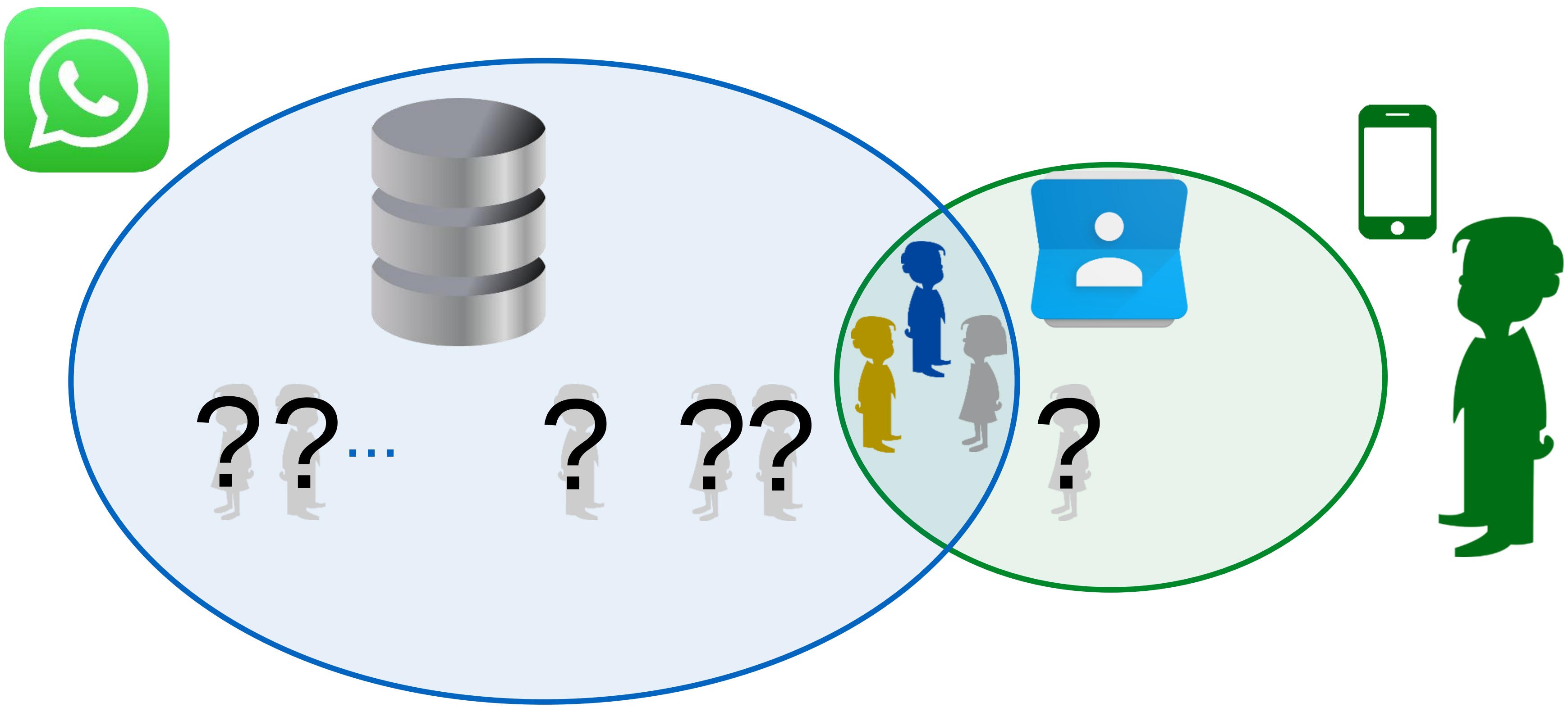


PSI with Unequal Set Sizes – Mobile Messaging Service



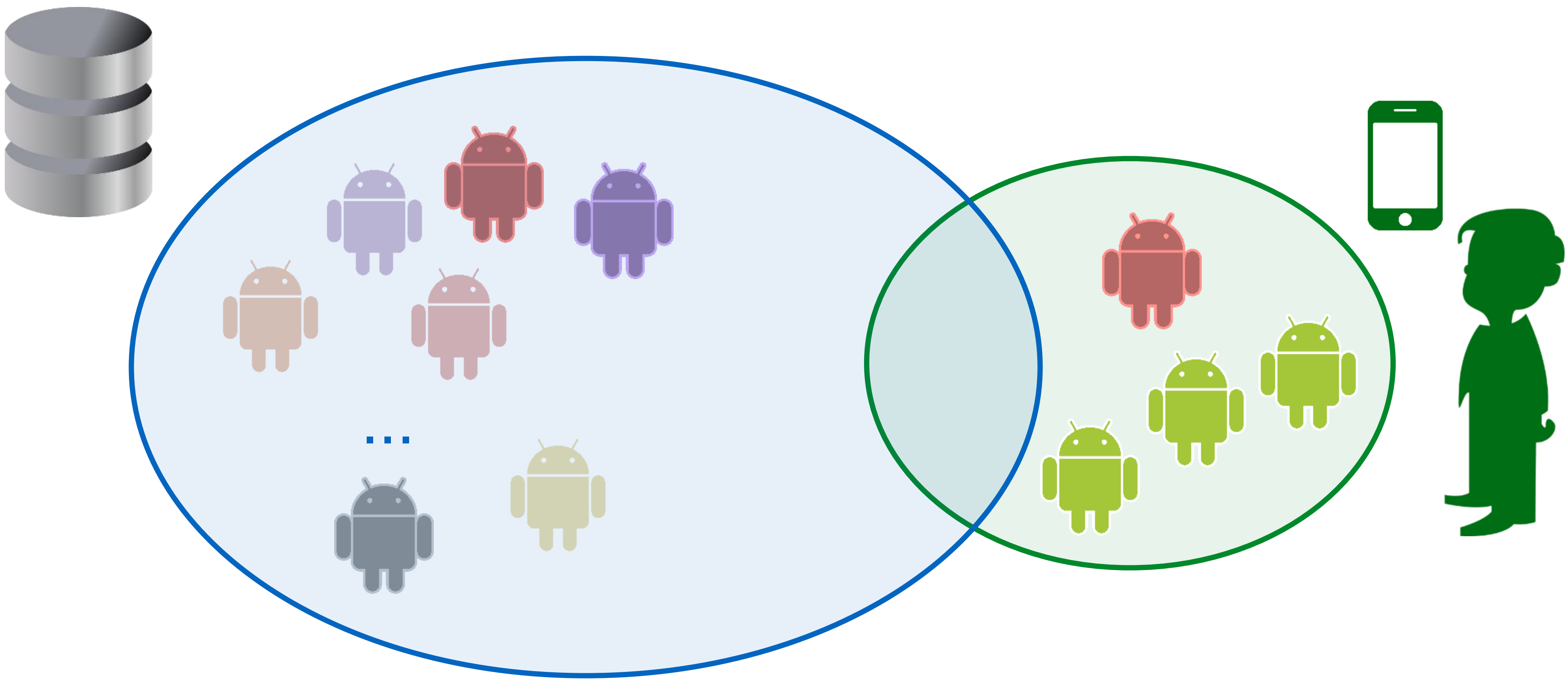
$$N_A \gg N_B$$

PSI with Unequal Set Sizes – Mobile Messaging Service



$$N_A \gg N_B$$

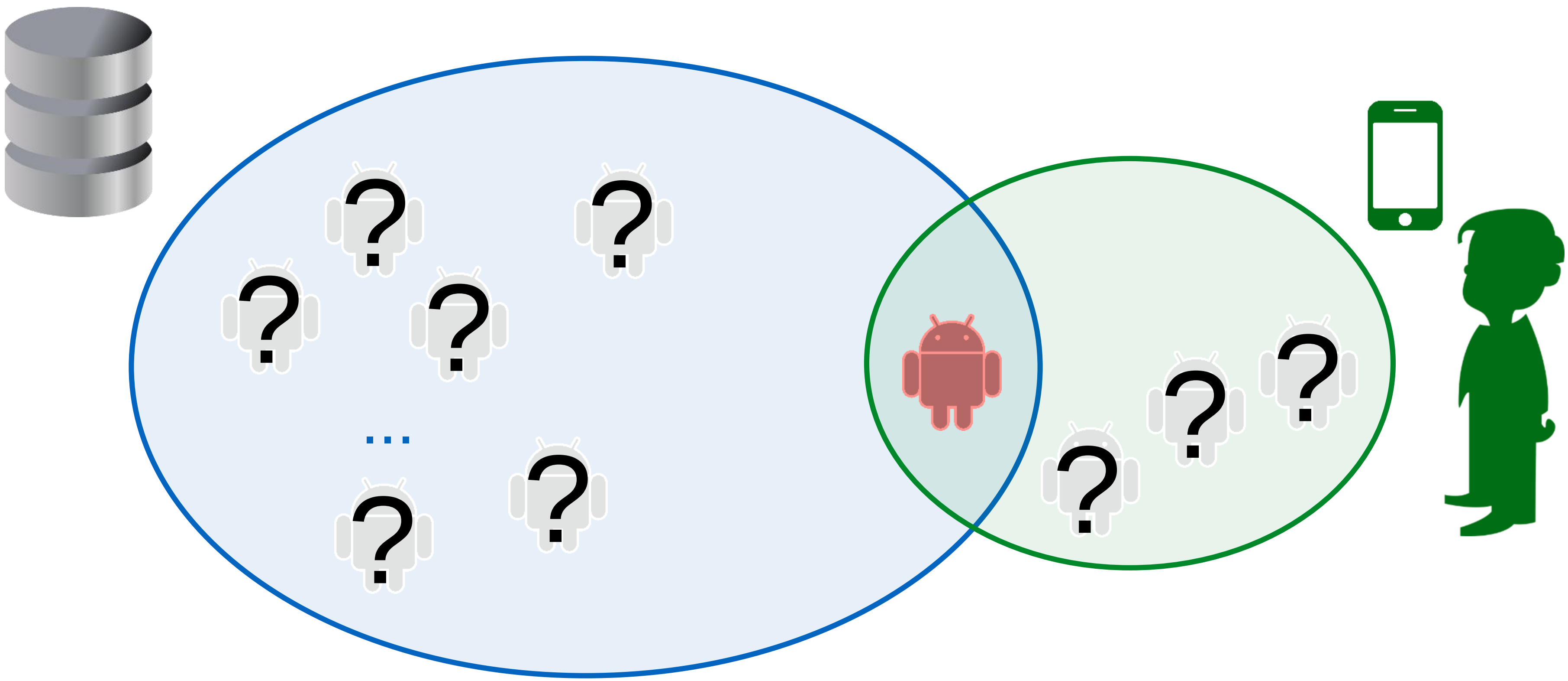
PSI with Unequal Set Sizes – Malware Detection Service



KASPERSKY N_A $\gg$ N_B

3 Mio 95 [TLP+17]

PSI with Unequal Set Sizes – Malware Detection Service



KASPERSKY

$N_A \gg N_B$

3 Mio | 95

[TLP+17]

What do we have?

- **OT-based protocols efficient for $N_A \approx N_B$**
  - Garbled BF based protocols [DCW13,RR17]
  - Hashing-based protocols [PSZ14,PSSZ15,KKRT16]

What do we have?

- **OT-based protocols efficient for $N_A \approx N_B$**
  - Garbled BF based protocols [DCW13,RR17]
  - Hashing-based protocols [PSZ14,PSSZ15,KKRT16]

Require sending data linear in N_A for each element of the client ($O(N_A N_B)$)

What do we have?

- **OT-based protocols efficient for $N_A \approx N_B$**
  - Garbled BF based protocols [DCW13,RR17]
  - Hashing-based protocols [PSZ14,PSSZ15,KKRT16]

Require sending data linear in N_A for each element of the client ($O(N_A N_B)$)

- **Protocols linear in the set sizes $(O(N_A + N_B))$**
  - Based on public-key crypto: OPE [FNP04], DH [HFH99]
  - Based on Oblivious PRF evaluation: NR [FIPR05,HL08], AES [PSSW09], RSA [CT10]

What do we have?

- **OT-based protocols efficient for $N_A \approx N_B$**
  - Garbled BF based protocols [DCW13,RR17]
  - Hashing-based protocols [PSZ14,PSSZ15,KKRT16]

Require sending data linear in N_A for each element of the client $(O(N_A N_B))$

- **Protocols linear in the set sizes $(O(N_A + N_B))$**
  - Based on public-key crypto: OPE [FNP04], DH [HFH99]
  - Based on Oblivious PRF evaluation: NR [FIPR05,HL08], AES [PSSW09], RSA [CT10]

Can these be adapted to unequal set sizes?

What do we have?

- **OT-based protocols efficient for $N_A \approx N_B$**
  - Garbled BF based protocols [DCW13,RR17]
  - Hashing-based protocols [PSZ14,PSSZ15,KKRT16]

Require sending data linear in N_A for each element of the client ($O(N_A N_B)$)

- **Protocols linear in the set sizes $(O(N_A + N_B))$**
  - Based on public-key crypto: OPE [FNP04], DH [HFH99]
  - Based on Oblivious PRF evaluation: NR [FIPR05,HL08], AES [PSSW09], RSA [CT10]

Can these be adapted to unequal set sizes?

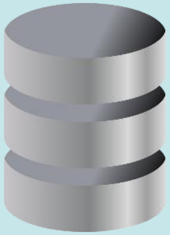
Our Contributions

Improve existing linear complexity protocols for unequal set sizes

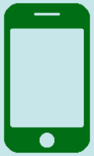
Prototype implementation of improved protocols

Further extensions for real-world applications

Precomputed PSI – Three Phases



$|N_A| \gg |N_B|$



Base Phase

Data-independent, depends on $N_B^{\max}$ maximum number of client inputs

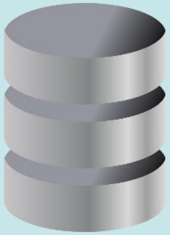
Setup Phase

Depends on the N_A elements in the database

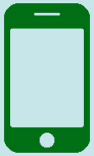
Online Phase

Depends on the N_B elements in the client set

Precomputed PSI – Three Phases



$|N_A| \gg |N_B|$



Base Phase

Data-independent, depends on $N_B^{\max}$ maximum number of client inputs

Can be precomputed without any knowledge on the inputs

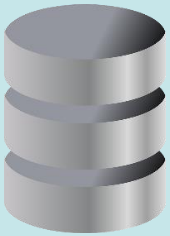
Setup Phase

Depends on the N_A elements in the database

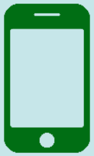
Online Phase

Depends on the N_B elements in the client set

Precomputed PSI – Three Phases



$|N_A| \gg |N_B|$



Base Phase

Data-independent, depends on $N_B^{\max}$ maximum number of client inputs

Can be precomputed without any knowledge on the inputs

Setup Phase

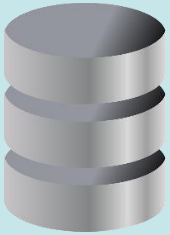
Depends on the N_A elements in the database

The server can perform most of the computation in advance

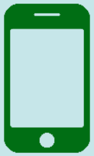
Online Phase

Depends on the N_B elements in the client set

Precomputed PSI – Three Phases



$|N_A| \gg |N_B|$



Base Phase

Data-independent, depends on $N_B^{\max}$ maximum number of client inputs

Can be precomputed without any knowledge on the inputs

Setup Phase

Depends on the N_A elements in the database

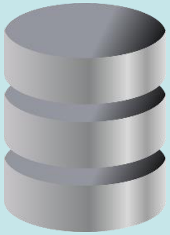
The server can perform most of the computation in advance

Same for all clients?

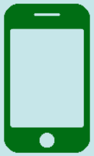
Online Phase

Depends on the N_B elements in the client set

Precomputed PSI – Three Phases



$|N_A| \gg |N_B|$



Base Phase

Data-independent, depends on $N_B^{\max}$ maximum number of client inputs

Can be precomputed without any knowledge on the inputs

Setup Phase

Depends on the N_A elements in the database

The server can perform most of the computation in advance

Same for all clients?

Online Phase

Depends on the N_B elements in the client set

Computation on the client's few elements is fast

Bloom filter

H_1 H_2 ... H_k

1	2	...	*i*		...			*j*			...			*n*
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bloom filter

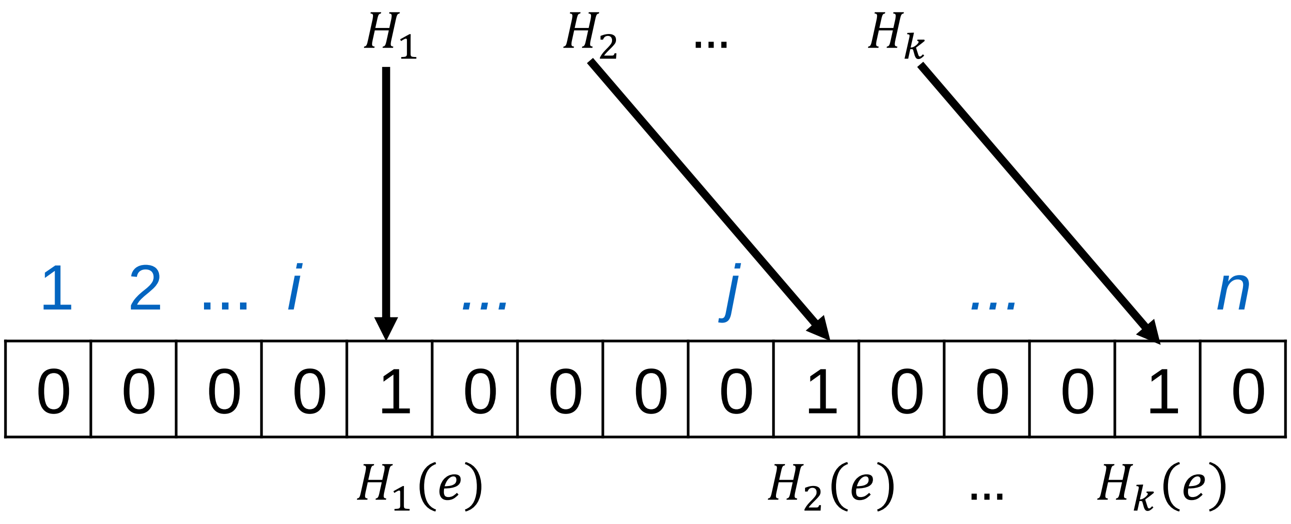
e: 004912345678910

H_1 H_2 ... H_k

1	2	...	i		...			j			...			n
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

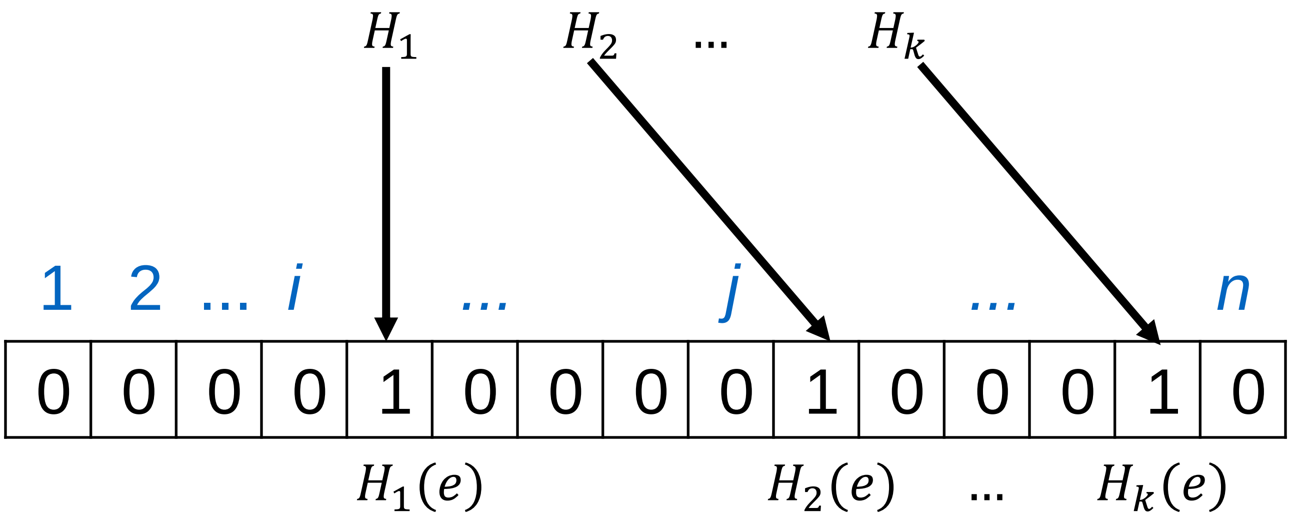
Bloom filter

e: 004912345678910



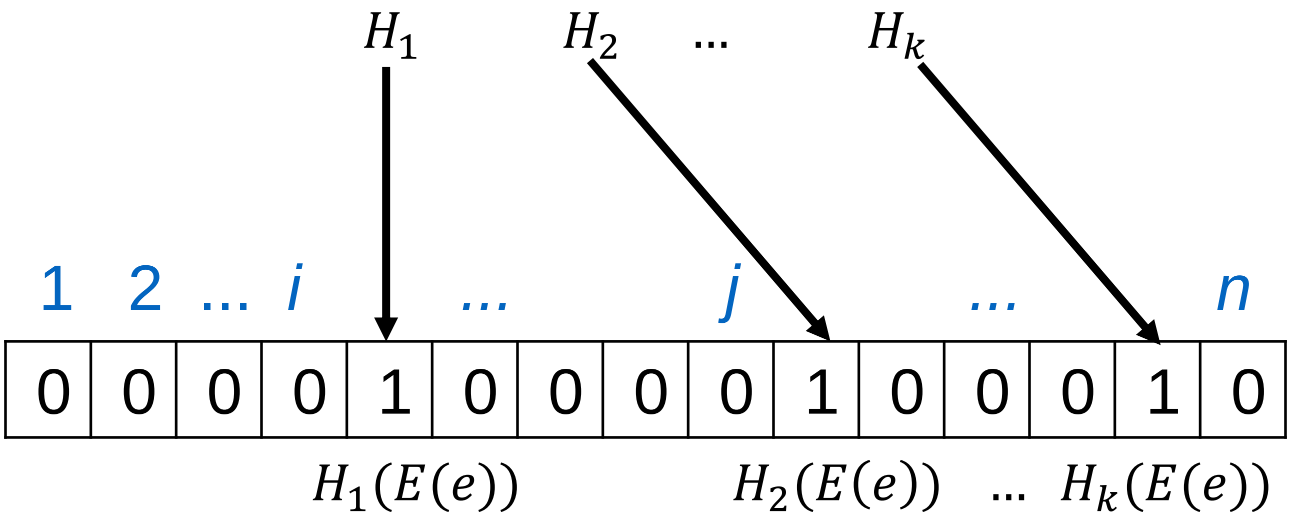
Bloom filter

e: 004912345678910



Bloom filter

$E(e)$: fti45jxcfuu984fghdr56fguew91jm



Efficient and Secure Updates

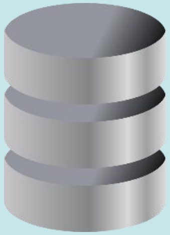
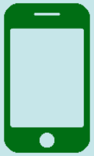
Insertion in Bloom filter

$E(e)$: fti45jxcfuu984fghdr56fguew91jm

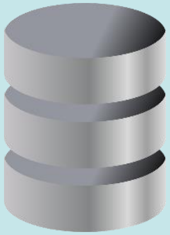
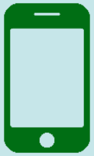
$$H_1(E(e)), H_2(E(e)), \dots, H_k(E(e))$$

Deletion: Counting Bloom filter

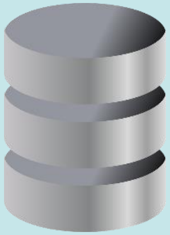
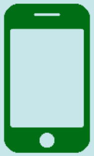
Precomputed PSI – PSI with PRF: RSA-PSI, NR-PSI, GC-PSI

 $\lvert N_A \rvert$	$\gg$	$\lvert N_B \rvert$ 
Key generation, Precomputation	**Base Phase** Agree on key, parameters, Precomputation	Precomputation
Encrypt database and insert in Bloom filter	**Setup Phase** BF →	Store Bloom filter
Encrypt element privately	**Online Phase** ← Encryption →	Request encryption of elements privately Check and output intersection locally

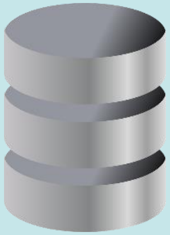
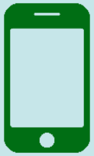
Precomputed PSI – PSI with PRF: RSA-PSI, NR-PSI, GC-PSI

 $\lvert N_A \rvert$	$\gg$	$\lvert N_B \rvert$ 
Key generation, Precomputation	**Base Phase** Agree on key, parameters, Precomputation	Precomputation
Encrypt database and insert in Bloom filter	**Setup Phase** BF →	Store Bloom filter
Encrypt element privately	**Online Phase** ← Encryption →	Request encryption of elements privately Check and output intersection locally

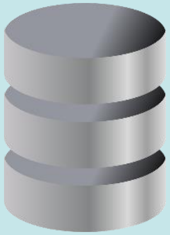
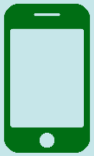
Precomputed PSI – PSI with PRF: RSA-PSI, NR-PSI, GC-PSI

 $\lvert N_A \rvert$	$\gg$	$\lvert N_B \rvert$ 
Key generation, Precomputation	**Base Phase** Agree on key, parameters, Precomputation	Precomputation
Encrypt database and insert in Bloom filter	**Setup Phase** BF →	Store Bloom filter
Encrypt element privately	**Online Phase** ← Encryption →	Request encryption of elements privately Check and output intersection locally

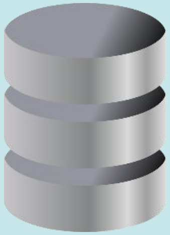
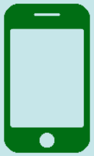
Precomputed PSI – PSI with Diffie-Hellman – DH-PSI

 $\lvert N_A \rvert$	$\gg$	$\lvert N_B \rvert$ 
Key generation α	**Base Phase**	Key generation β
Encrypt database with α	**Setup Phase** Encryption →	Encrypt encrypted database with β and insert in Bloom filter
Encrypt encrypted elements with α	**Online Phase** Encryption with β ← Encryption with $\alpha\beta$ →	Encrypt elements with β Check intersection

Precomputed PSI – PSI with Diffie-Hellman – DH-PSI

 $\lvert N_A \rvert$	$\gg$	$\lvert N_B \rvert$ 
Key generation α	**Base Phase**	Key generation β
Encrypt database with α	**Setup Phase** Encryption →	Encrypt encrypted database with β and insert in Bloom filter
Encrypt encrypted elements with α	**Online Phase** Encryption with β ← Encryption with $\alpha\beta$ →	Encrypt elements with β Check intersection

Precomputed PSI – PSI with Diffie-Hellman – DH-PSI

 $\lvert N_A \rvert$	$\gg$	$\lvert N_B \rvert$ 
Key generation α	**Base Phase**	Key generation β
Encrypt database with α	**Setup Phase** Encryption →	Encrypt encrypted database with β and insert in Bloom filter
Encrypt encrypted elements with α	**Online Phase** ← Encryption with β Encryption with $\alpha\beta$ →	Encrypt elements with β Check intersection

Our Contributions

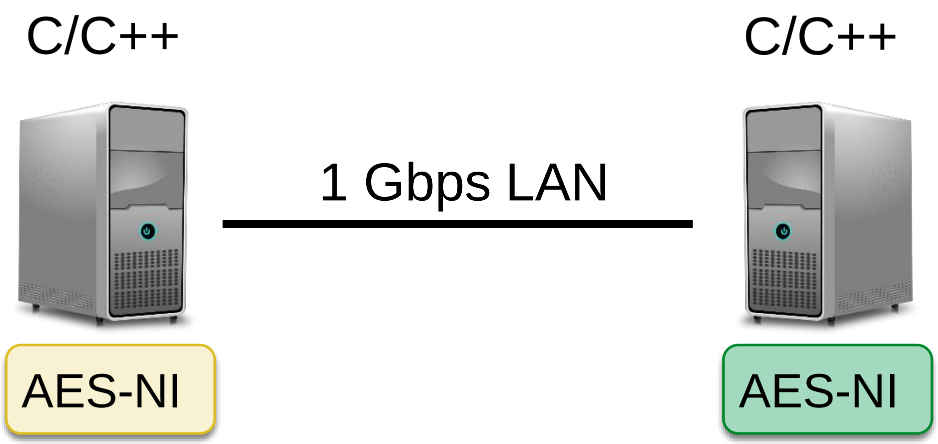
Improve existing linear complexity protocols for unequal set sizes

Prototype implementation of improved protocols

Further extensions for real-world applications

Computation and Communication – PC Malware Detection

$$\mathrm{FPR} = 10^{-3}$$
$$N_A = 2^{20}$$
$$N_B = 128$$
$$N_B^{\max} = 256$$



Protocol\Phase	Base phase		Setup phase		Online phase	
RSA-PSI	14 ms	0 MB	57.4 min	1.8 MB	0.9 sec	0.1 MB
ECC-DH-PSI	1 ms	0 MB	22.1 min	35.5 MB	0.4 sec	0.1 MB
NR-PSI	0.1 sec	2.2 MB	12.6 min	1.8 MB	1.4 sec	0.5 MB
AES-NI GC-PSI	1.3 sec	44.5 MB	0.3 sec	1.8 MB	0.3 sec	0.5 MB

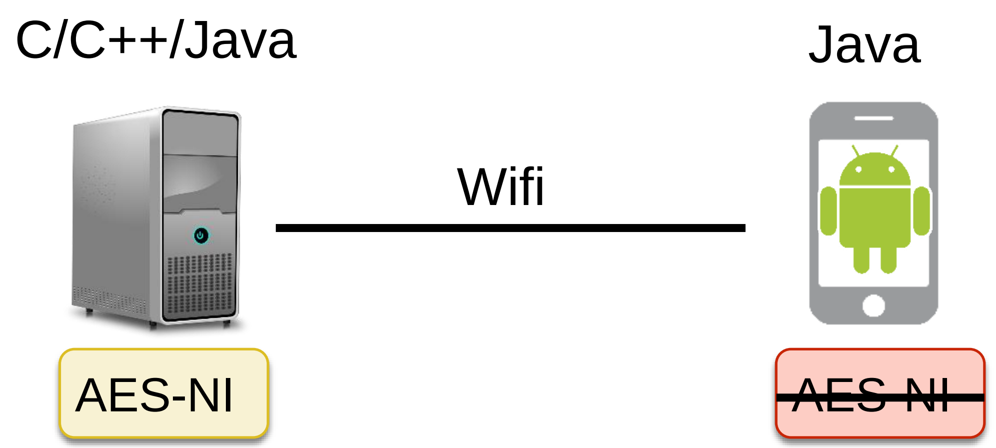
Computation and Communication – Mobile Malware Detectio

$\mathrm{FPR} = 10^{-3}$

$N_A = 2^{20}$

$N_B = 128$

$N_B^{\max} = 256$



	Protocol\Phase	Base phase		Setup phase		Online phase	
	RSA-PSI	1.4 sec	0 MB	57.4 min	1.8 MB	7.7 sec	0.1 MB
~~ECC~~	DH-PSI	1 ms	0 MB	8.6 min	35.5 MB	2.9 sec	0.1 MB
	NR-PSI	0.7 min	2.2 MB	12.7 min	1.8 MB	31.6 sec	0.5 MB
~~AES-NI~~	GC-PSI	7.6 min	44.5 MB	1.7 sec	1.8 MB	18.1 min	0.5 MB

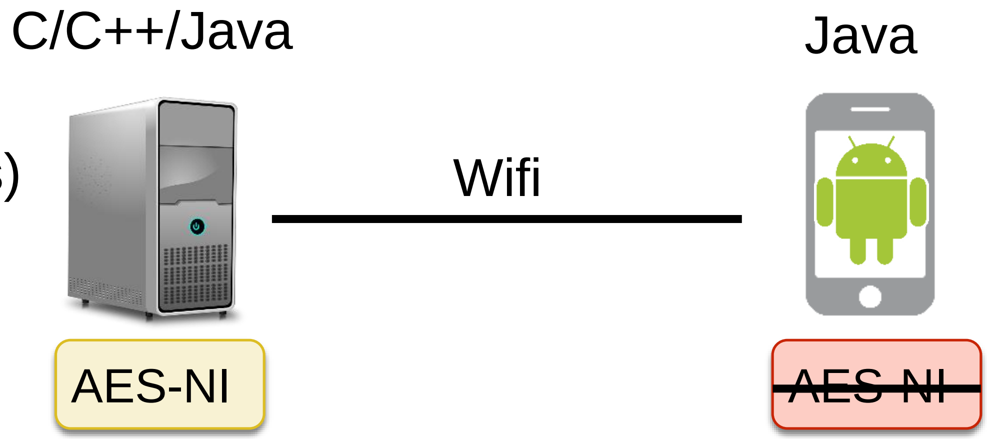
Computation and Communication– Mobile Messaging

$\mathrm{FPR} = 10^{-9}$

$N_A = 2^{30}$ (1 billion users)

$N_B = 256$

$N_B^{\max} = 512$



Protocol\Phase	Base phase		Setup phase		Online phase	
RSA-PSI	2.7 sec	0 MB	40.8 days	5.4 GB	15.4 sec	0.2 MB
~~ECC~~ DH-PSI	1 ms	0 MB	6.1 days	256 GB	5.9 sec	0.2 MB
NR-PSI	0.7 min	4.2 MB	9.0 days	5.4 GB	1.1 min	1.0 MB
~~AES-NI~~ GC-PSI	7.6 min	89.0 MB	0.5 hour	5.4 GB	0.6 hour	1.0 MB

Computation and Communication– Mobile Messaging

$\mathrm{FPR} = 10^{-9}$

$N_A = 2^{30}$ (1 billion users)

$N_B = 256$

$N_B^{\max} = 512$

C/C++/Java

AES-NI

Wifi

Java

~~AES-NI~~

Protocol\Phase	Base phase		Setup phase		Online phase	
RSA-PSI	2.7 sec	0 MB	40.8 days	5.4 GB	15.4 sec	0.2 MB
~~ECC~~ DH-PSI	1 ms	0 MB	6[illegible]s	256 GB	5.9 sec	0.2 MB
NR-PSI	0.7 min	4.2 MB	9.0 days	5.4 GB	1.1 min	1.0 MB
~~AES-NI~~ GC-PSI	7.6 min	89.0 MB	0.5 hour	5.4 GB	0.6 hour	1.0 MB

Our Contributions

Improve existing linear complexity protocols for unequal set sizes

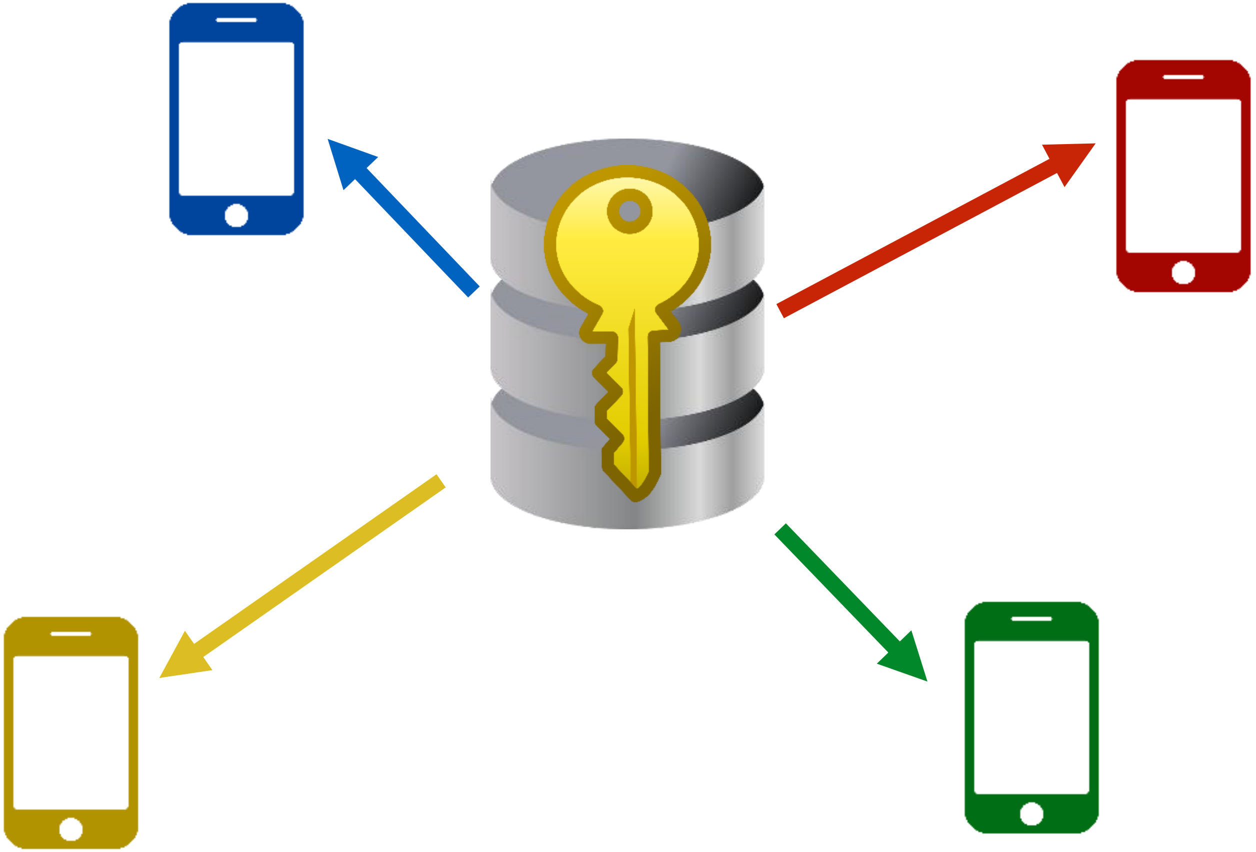
Prototype implementation of improved protocols

Further extensions for real-world applications

Same Encrypted Database for Multiple Clients

Same Encrypted Database for Multiple Clients

Content delivery network (CDN)



Effect on Performance – Mobile Messaging

$\text{FPR} = 10^{-9}$

$N_A = 2^{30}$ (1 billion users)

$N_B = 512$

$N_B^{\text{max}} = 512$

Protocol\Phase	Base phase		Setup phase		Online phase	
RSA-PSI	2.7 sec	0 MB	~~40.8 days~~	5.4 GB	30.7 sec	0.3 MB
DH-PSI	1 ms	0 MB	6.1 days	256 GB	11.8 sec	0.3 MB
NR-PSI	0.7 min	4.2 MB	~~9.0 days~~	5.4 GB	2.1 min	2.0 MB
GC-PSI	7.6 min	89.0 MB	~~0.5 hours~~	5.4 GB	1.2 hours	2.0 MB

Effect on Performance – Mobile Messaging

$\mathrm{FPR} = 10^{-9}$

$N_A = 2^{30}$ (1 billion users)

$N_B = 512$

$N_B^{\max} = 512$

Protocol\Phase	Base phase		Setup phase		Online phase	
RSA-PSI	2.7 sec	0 MB	~~40.8 days~~	5.4 GB	30.7 sec	0.3 MB
~~DH-PSI~~	~~1 ms~~	~~0 MB~~	~~6.1 days~~	~~256 GB~~	~~11.8 sec~~	~~0.3 MB~~
NR-PSI	0.7 min	4.2 MB	~~9.0 days~~	5.4 GB	2.1 min	2.0 MB
GC-PSI	7.6 min	89.0 MB	~~0.5 hours~~	5.4 GB	1.2 hours	2.0 MB

Effect on Performance – Mobile Messaging

$\mathrm{FPR} = 10^{-9}$

$N_A = 2^{30}$ (1 billion users)

$N_B = 512$

$N_B^{\max} = 512$

Protocol\Phase	Base phase		Setup phase		Online phase	
RSA-PSI	2.7 sec	0 MB	40.8 days	5.4 GB	30.7 sec	0.3 MB
DH-PSI	1 ms	0 MB	6.1 days	256 GB	11.8 sec	0.3 MB
NR-PSI	0.7 min	4.2 MB	9.0 days	5.4 GB	2.1 min	2.0 MB
GC-PSI	7.6 min	89.0 MB	0.5 hours	5.4 GB	1.2 hours	2.0 MB

Effect on Performance – Mobile Messaging

$\mathrm{FPR} = 10^{-9}$

$N_A = 2^{30}$ (1 billion users)

$N_B = 512$

$N_B^{\max} = 512$

Cuckoo filter in follow up work [RA17] → 4 GB

Protocol\Phase	Base phase		Setup phase		Online phase	
RSA-PSI	2.7 sec	0 MB	~~40.8 days~~	5.4 GB	30.7 sec	0.3 MB
~~DH-PSI~~	~~1 ms~~	~~0 MB~~	~~6.1 days~~	~~256 GB~~	~~11.8 sec~~	~~0.3 MB~~
NR-PSI	0.7 min	4.2 MB	~~9.0 days~~	5.4 GB	2.1 min	2.0 MB
GC-PSI	7.6 min	89.0 MB	~~0.5 hours~~	5.4 GB	1.2 hours	2.0 MB

Summary

Improve existing linear complexity protocols for unequal set sizes

Prototype implementation of improved protocols

Further extensions for real-world applications

Thank you for your attention!

References

[CT10]: E. De Cristofaro, G. Tsudik: *Practical private set intersection protocols with linear complexity*. In FC'10.

[DCW13]: C. Dong, L. Chen, Z. Wen: *When private set intersection meets big data: an efficient and scalable protocol.* In CCS'13.

[FIPR05]: M. J. Freedman, Y. Ishai, B. Pinkas, O. Reingold: *Keyword search and oblivious pseudorandom functions.* In TCC'05.

[FNP04]: M. J. Freedman, K. Nissim, B. Pinkas: *Efficient private matching and set intersection*. In Eurocrypt'04.

[HFH99]: B. A. Huberman, M. K. Franklin, T. Hogg: *Enhancing privacy and trust in electronic communities.* In EC'99.

[HL08]: C. Hazay, Y. Lindell: *Efficient protocols for set intersection and pattern matching with security against malicious adversaries.* In TCC'08.

[KKRT16]: V. Kolesnikov, R. Kumaresan, M. Rosulek, N. Trieu: *Efficient batched oblivious PRF with applications to private set intersection.* In CCS'16.

References

[PSSW09]: B. Pinkas, T. Schneider, N. P. Smart, S. C. Williams: *Secure two-party computation is practical.* In Asiacrypt'09.

[PSSZ15]: B. Pinkas, T. Schneider, G. Segev, M. Zohner: *Phasing: Private set intersection using permutation-based hashing.* In USENIX Security'15.

[PSZ14]: B. Pinkas, T. Schneider, M. Zohner: *Faster private set intersection based on OT extension.* In USENIX Security'14.

[RA17]: A. C. D. Resende, D. F. Aranha: *Unbalanced Approximate Private Set Intersection.* Eprint 2017/677.

[RR17]: P. Rindal, M. Rosulek: *Improved private set intersection against malicious adversaries.* In Eurocrypt'17.

[TLP+17]: S. Tamrakar, J. Liu, A. Paverd, J. Ekberg, B. Pinkas, N. Asokan: *The circle game: Scalable private membership test using trusted hardware.* In AsiaCCS'17.