Arke

Privacy-preserving and Decentralised Contact Discovery

Nicolas Mohnblatt | 10 Apr 2024

joint work with: **Alberto Sonnino, Philipp Jovanovic and Kobi Gurkan**

GR geometry RESEARCH

This Talk

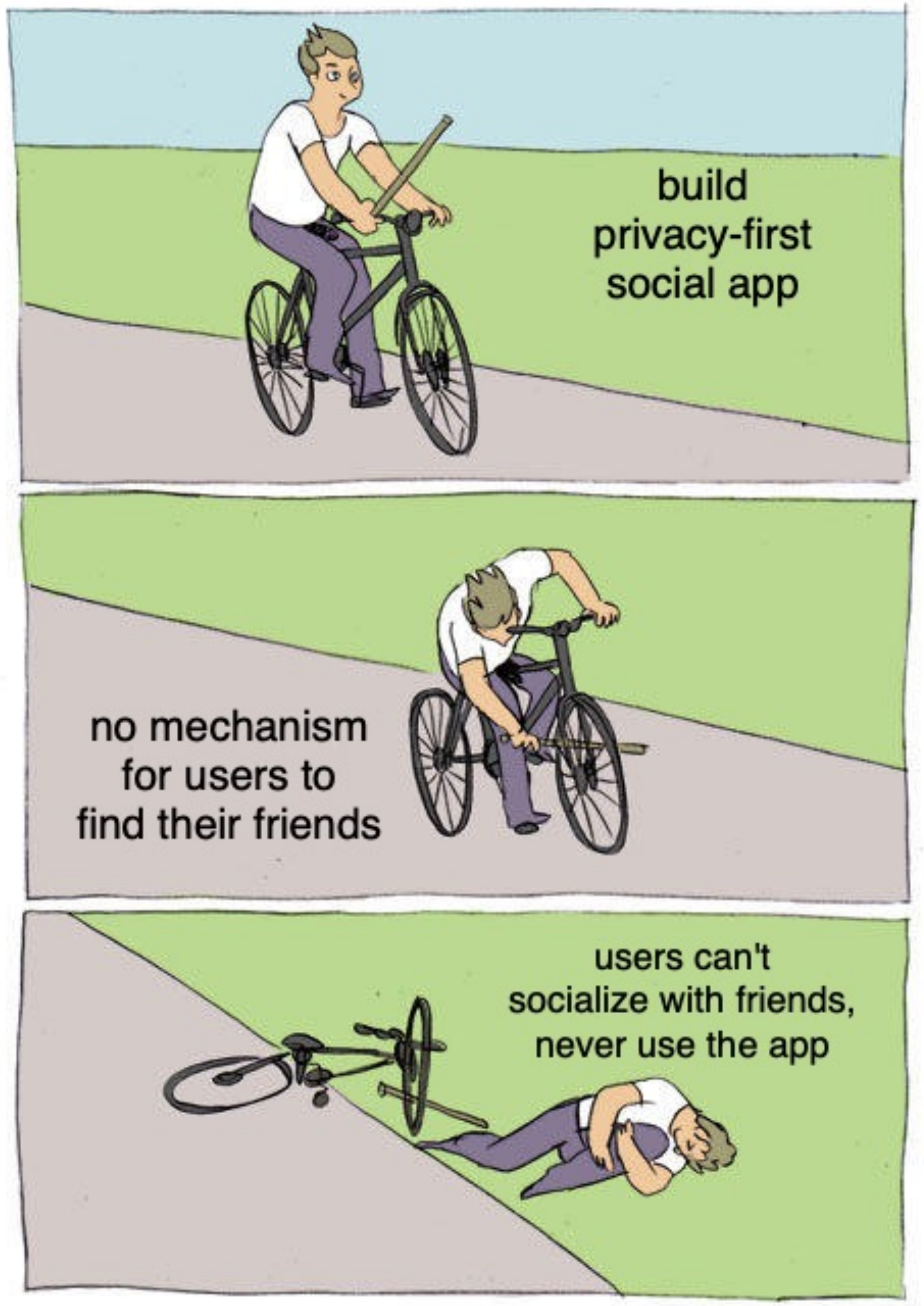
Using *identity-based cryptography* to bootstrap decentralised social apps

- Motivation: contact discovery, privacy, threat model
- Good attempts that fall short
- *Arke* pt. 1: high-level construction
- *Arke* pt. 2: cryptography 🧑‍🍳
- Other applications and warnings

The year is ~~202X~~ Today!, we can:

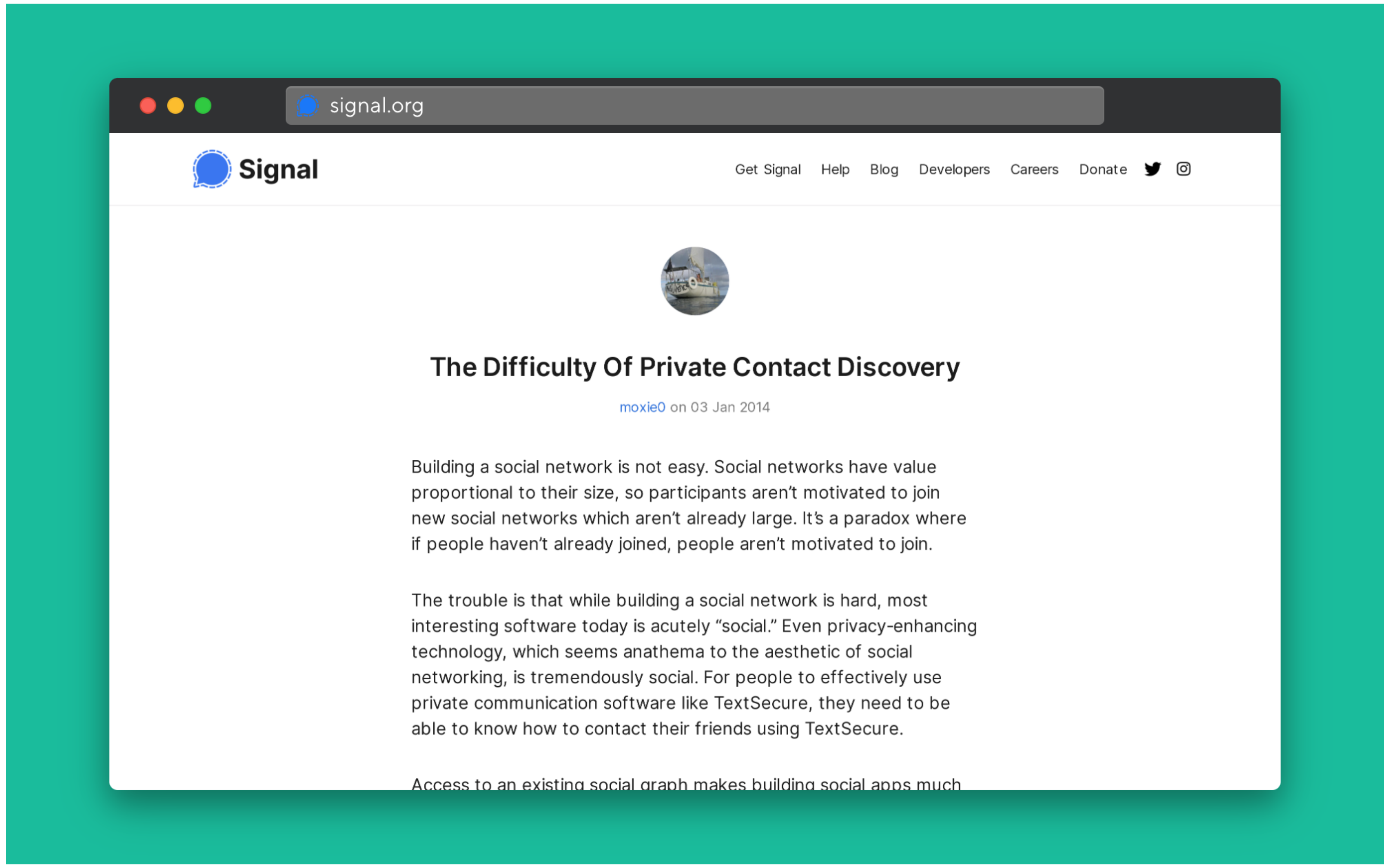
- create anonymous blockchain accounts from our email address.
- zk-prove ownership of social media handles.
- can build end-to-end private messaging and payments apps.

Ready to build the next generation of social apps?


build privacy-first social app
no mechanism for users to find their friends
users can't socialize with friends, never use the app

Ready to build the next generation of social apps?

- Moxie Marlinspike (Signal), 2014:
  - “most interesting software today is acutely ‘social’. Even privacy-enhancing technology […] is tremendously social.”
  - “networks have value proportional to their size”
  - “**Access to an existing social graph** makes building social apps much easier”



The Difficulty Of Private Contact Discovery

moxie0 on 03 Jan 2014

Building a social network is not easy. Social networks have value proportional to their size, so participants aren’t motivated to join new social networks which aren’t already large. It’s a paradox where if people haven’t already joined, people aren’t motivated to join.

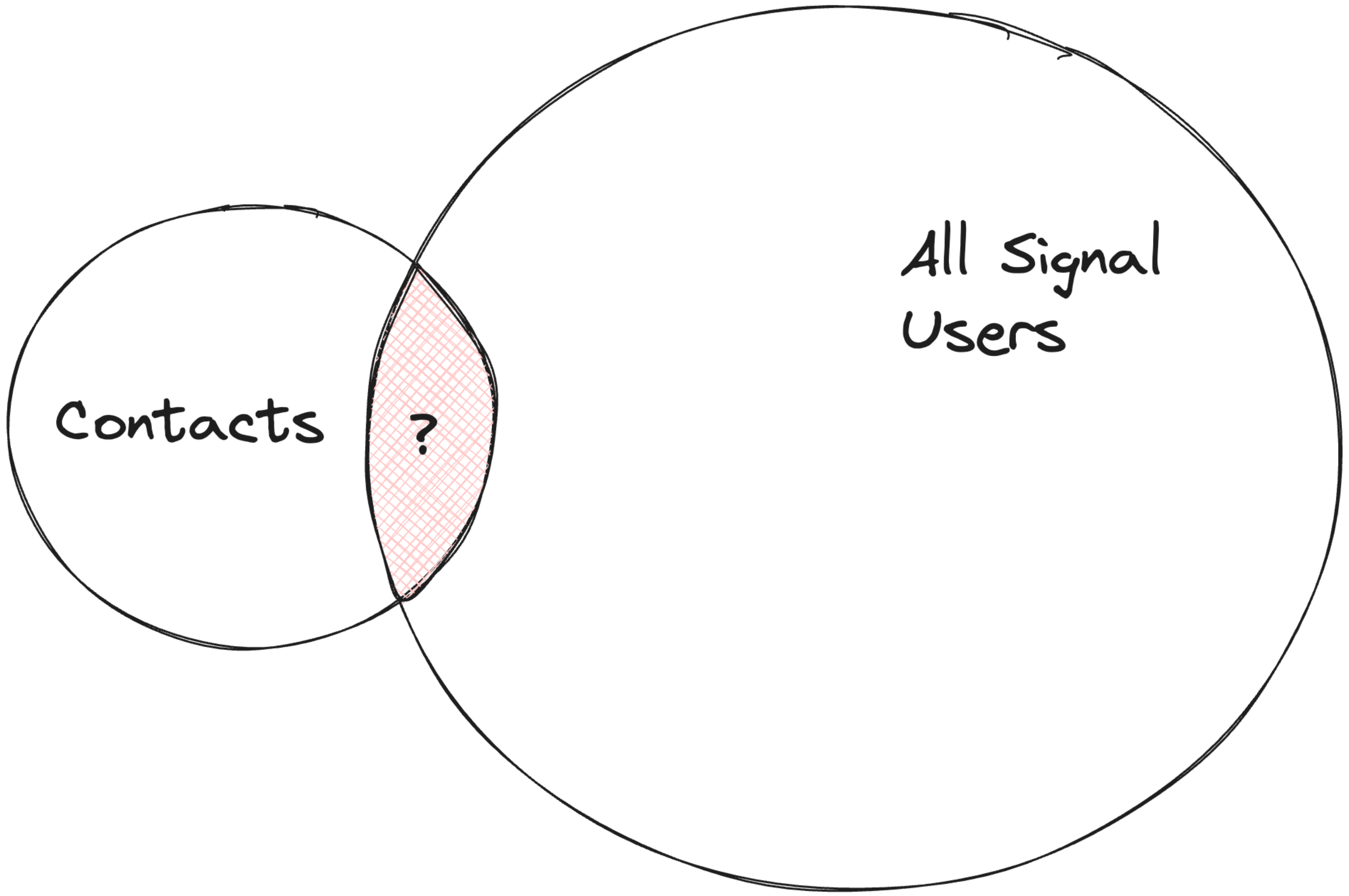
The trouble is that while building a social network is hard, most interesting software today is acutely “social.” Even privacy-enhancing technology, which seems anathema to the aesthetic of social networking, is tremendously social. For people to effectively use private communication software like TextSecure, they need to be able to know how to contact their friends using TextSecure.

Access to an existing social graph makes building social apps much

Contact Discovery

Accessing an existing social graph

- Consider a user joining Signal:
  - they want to know who in their contacts also uses Signal
- Also need additional information: where to send messages, what is my friend’s public key, etc…
- Can extend this to other social graphs:
  - email, X followers, etc…

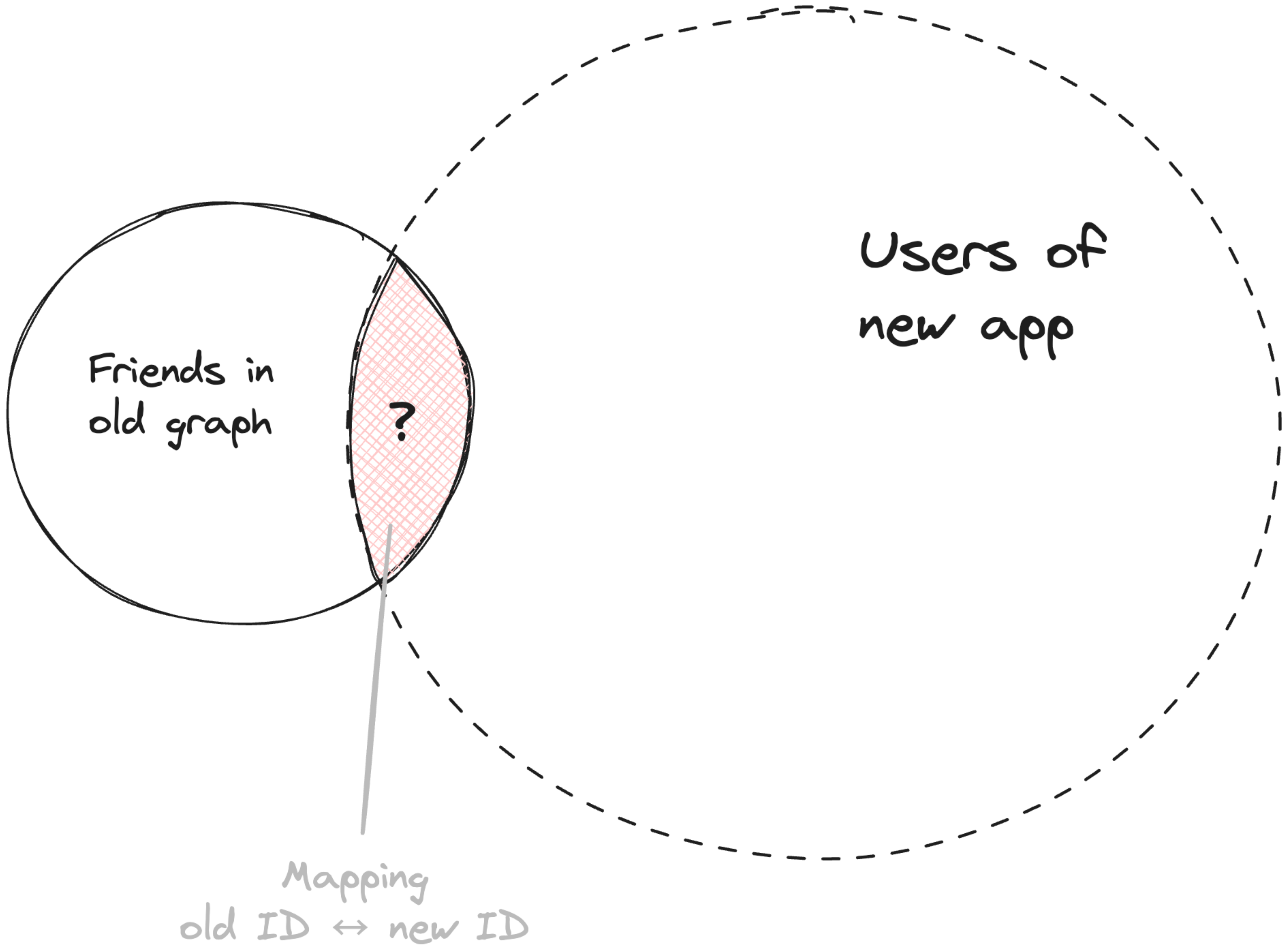


Contact Discovery

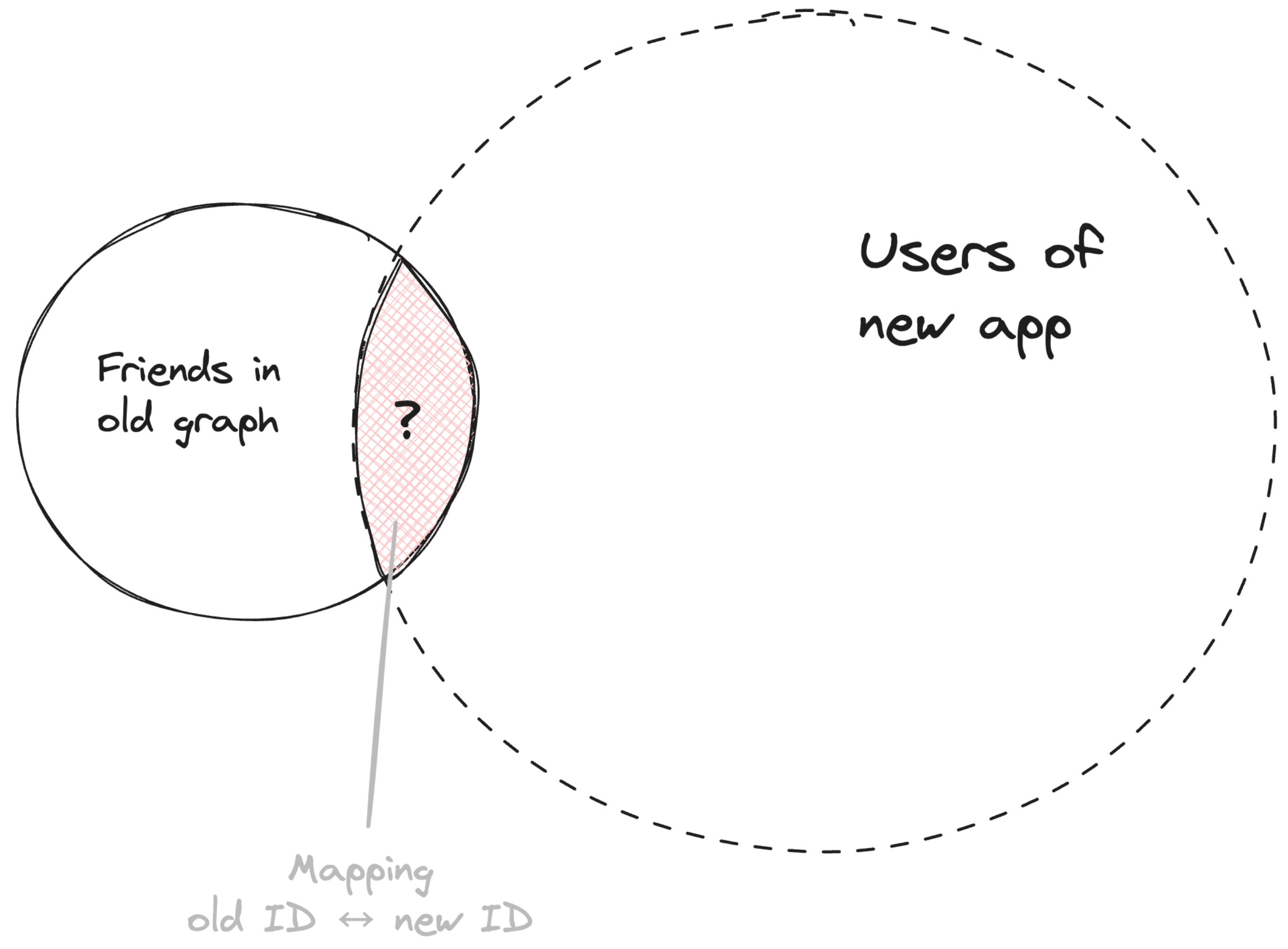
Accessing an existing social graph

- Consider a user joining Signal:
  - they want to know who in their contacts also uses Signal
- Also need additional information: where to send messages, what is my friend's public key, etc…
- Can extend this to other social graphs:
  - email, X followers, etc…

User	existing ID	New ID
Alice	@alice	0xA68E...F9
Bob	@bob	0xB935...CE



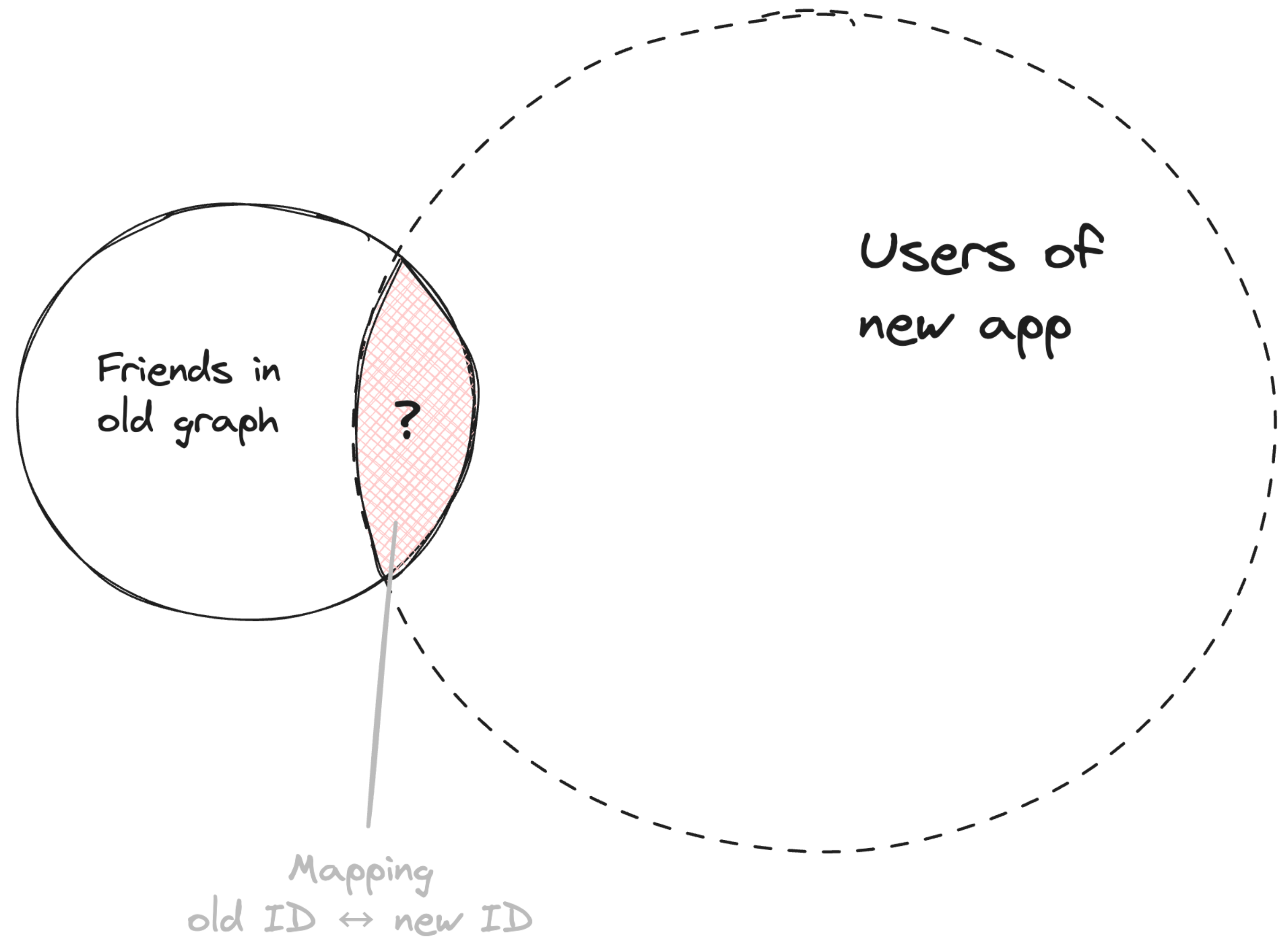
Private Contact Discovery



- Privacy means:
    1. nobody learns my contacts.
    2. I can filter who learns my “old ID ↔ new ID” correspondence.
    3. (hard mode) non-friends cannot know whether or not I am on the new app.
- Against anyone: external observers, (dis)honest users, the service itself.

Existing Approaches

Related Work



Approach	Description	Pros	Cons
Basic	upload contacts to a remote server (WhatsApp, Telegram)	works	no privacy
Private Set Intersection (PSI)	• user and server jointly compute the "intersection" • uses FHE (Kiss et al., Kales et al.) or TEE+oblivious RAM (Signal)	contacts remain private	• requires one entity to know: set of all users, and ID mapping • crawling/scraping attacks (Hagen et al., Facebook scrape) • (trusted hardware)
Peer-to-peer (P2P)	users send new ID as a secret message to their friends, and only their friends! (our approach + concurrent works UDM and Pudding)	can achieve all our privacy goals	more work for each user

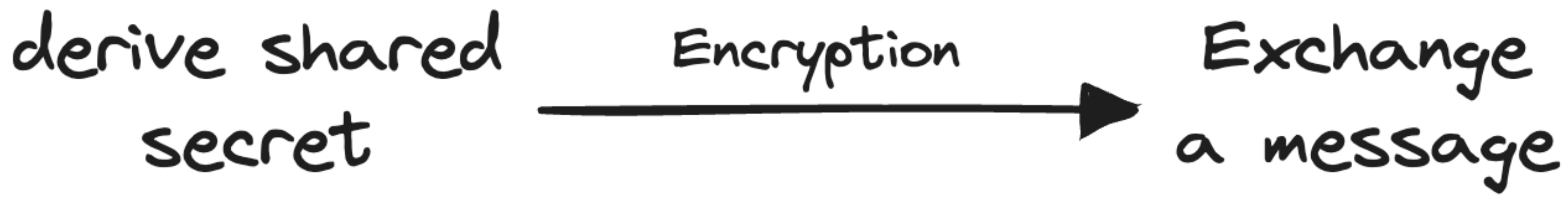
Peer-to-peer Contact Discovery

Users know their
friends' identifier

Exchange
a message

Peer-to-peer Contact Discovery

Users know their friends' identifier

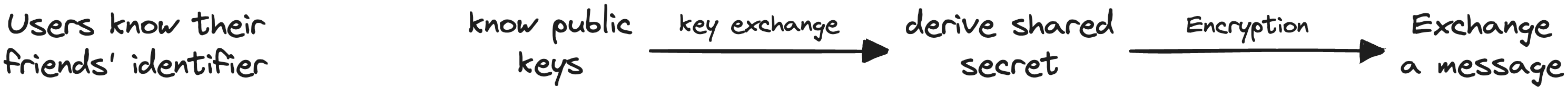


derive shared secret

Encryption

Exchange a message

Peer-to-peer Contact Discovery



Users know their friends' identifier

know public keys

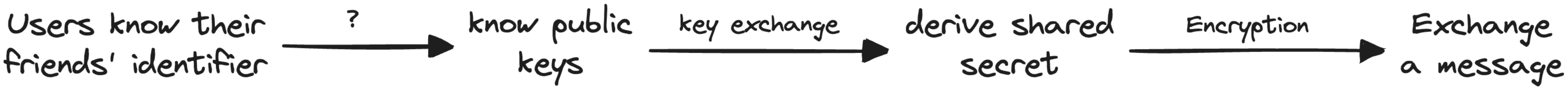
key exchange

derive shared secret

Encryption

Exchange a message

Peer-to-peer Contact Discovery



Users know their friends' identifier

?

know public keys

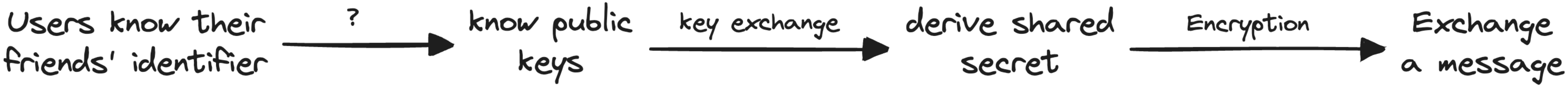
key exchange

derive shared secret

Encryption

Exchange a message

Peer-to-peer Contact Discovery



Users know their friends' identifier

?

know public keys

key exchange

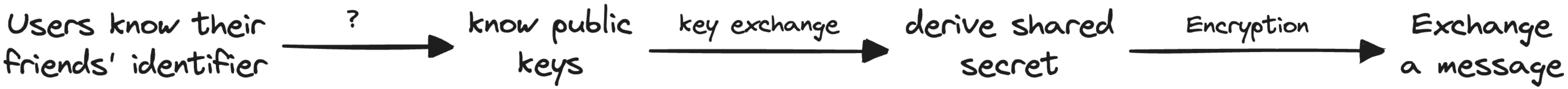
derive shared secret

Encryption

Exchange a message

identifier IS the public key !

Peer-to-peer Contact Discovery

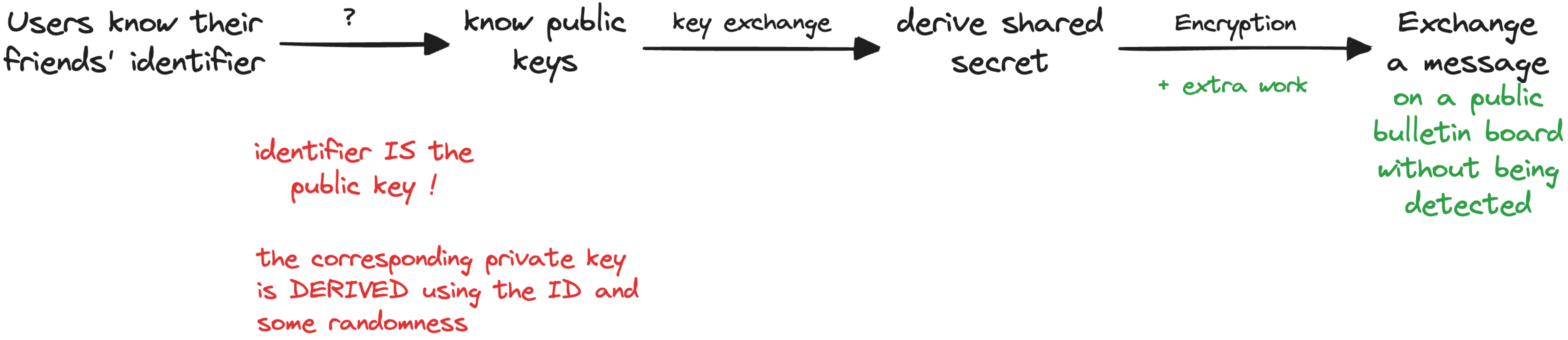


Users know their friends' identifier $\xrightarrow{?}$ know public keys $\xrightarrow{\text{key exchange}}$ derive shared secret $\xrightarrow{\text{Encryption}}$ Exchange a message

identifier IS the
public key !

the corresponding private key
is DERIVED using the ID and
some randomness

Peer-to-peer Contact Discovery



Users know their friends' identifier

?

know public keys

key exchange

derive shared secret

Encryption

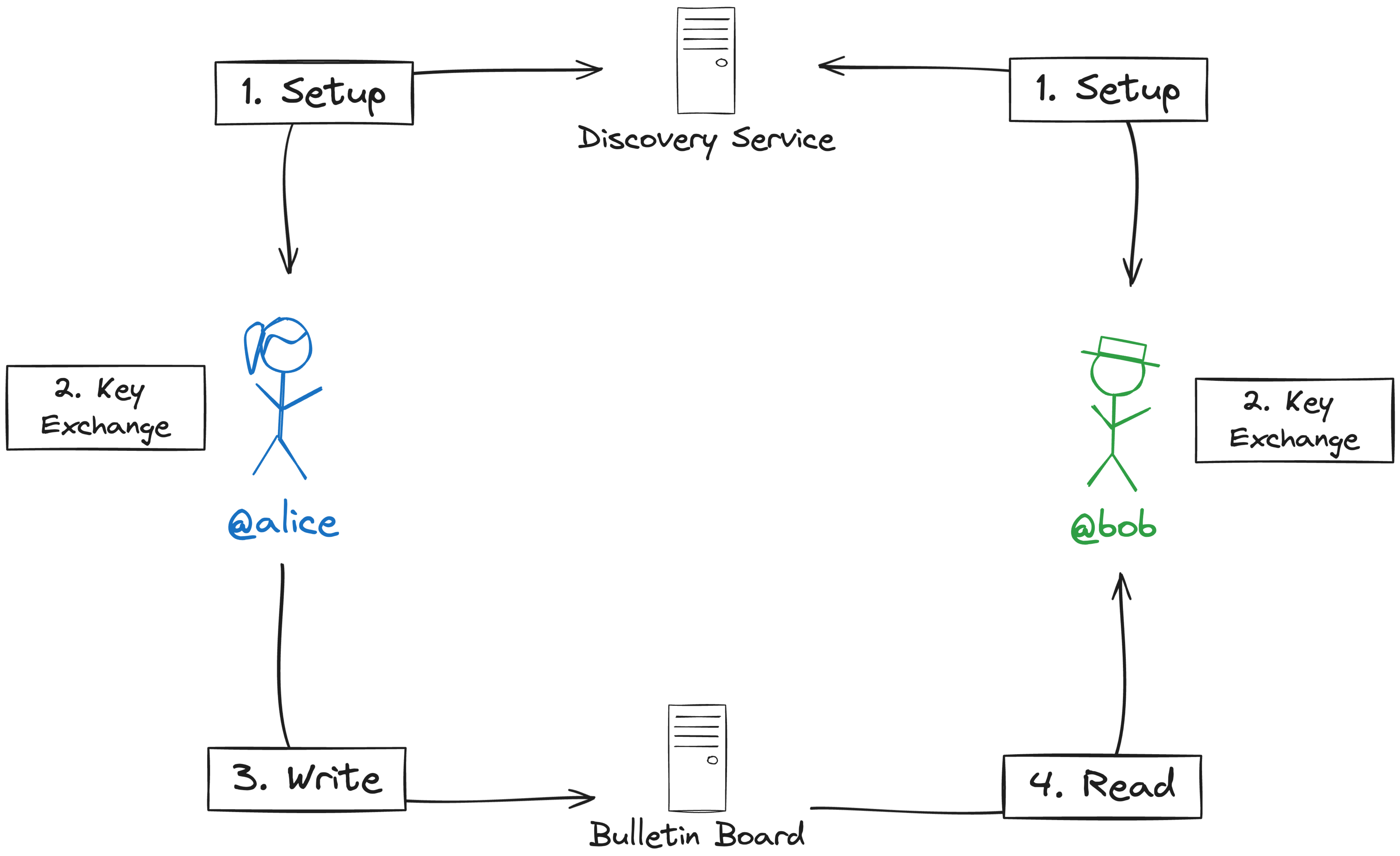
+ extra work

Exchange a message on a public bulletin board without being detected

identifier IS the public key !

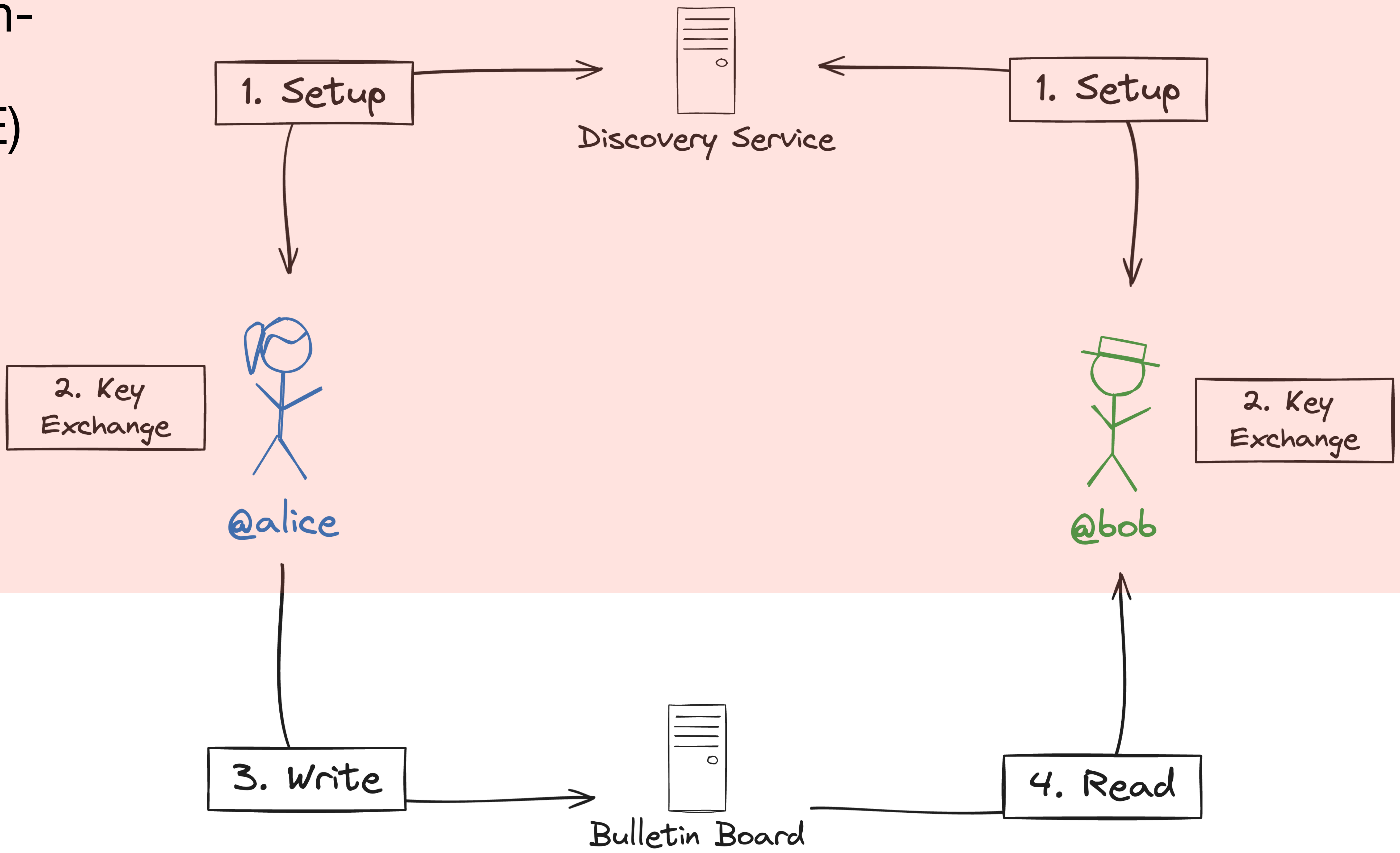
the corresponding private key is DERIVED using the ID and some randomness

Arke, System Overview



Arke, System Overview

IDentity-based Non-Interactive Key Exchange (ID-NIKE)



ID-NIKE - Semantics

Assuming a TTP

- Formally an ID-NIKE is defined by 3 algorithms:
  - $\mathsf{Setup}(\lambda) \to (\mathsf{pp}, \mathsf{msk})$
  - $\mathsf{Extract}(\mathsf{msk}, \mathsf{id}) \to sk_{\mathsf{id}}$
  - $\mathsf{SharedKey}(\mathsf{pp}, sk_{\mathsf{id}}, \mathsf{id}') \to k_{\mathsf{id},\mathsf{id}'}$
- Important that:

$$\mathsf{SharedKey}(\mathsf{pp}, sk_{\mathsf{id}}, \mathsf{id}') = \mathsf{SharedKey}(\mathsf{pp}, sk_{\mathsf{id}'}, \mathsf{id})$$

- Security: $k_{\mathsf{id},\mathsf{id}'}$ is *indistinguishable* from random.

ID-NIKE - Sakai-Ohgishi-Kasahara

Assuming a TTP

- Reminder (symmetric) pairings:

$$e\left(g^{a}, h^{b}\right) = e\left(h^{b}, g^{a}\right) = e\left(g, h\right)^{ab}$$

ID-NIKE - Sakai-Ohgishi-Kasahara

Assuming a TTP

$$e\left(g^{a}, h^{b}\right) = e\left(h^{b}, g^{a}\right) = e\left(g, h\right)^{ab}$$

- The SOK ID-NIKE:
  - $\mathsf{Setup}(\lambda) \rightarrow (\mathsf{pp} = g^{\mathsf{msk}}, \mathsf{msk} \leftarrow_{\$} \mathbb{Z}_p)$
  - $\mathsf{Extract}(\mathsf{msk}, \mathsf{id}) \rightarrow sk_{\mathsf{id}} = H(\mathsf{id})^{\mathsf{msk}}$
  - $\mathsf{SharedKey}(\mathsf{pp}, sk_{\mathsf{id}}, \mathsf{id}') \rightarrow k_{\mathsf{id},\mathsf{id}'} = e\left(sk_{\mathsf{id}}, H(\mathsf{id}')\right)$
- As required:

$$k_{\mathsf{id},\mathsf{id}'} = e\left(sk_{\mathsf{id}}, H(\mathsf{id}')\right) = e\left(H(\mathsf{id}), H(\mathsf{id}')\right)^{\mathsf{msk}} = e\left(sk_{\mathsf{id}'}, H(\mathsf{id})\right)$$

ID-NIKE - Sakai-Ohgishi-Kasahara

Removing the TTP

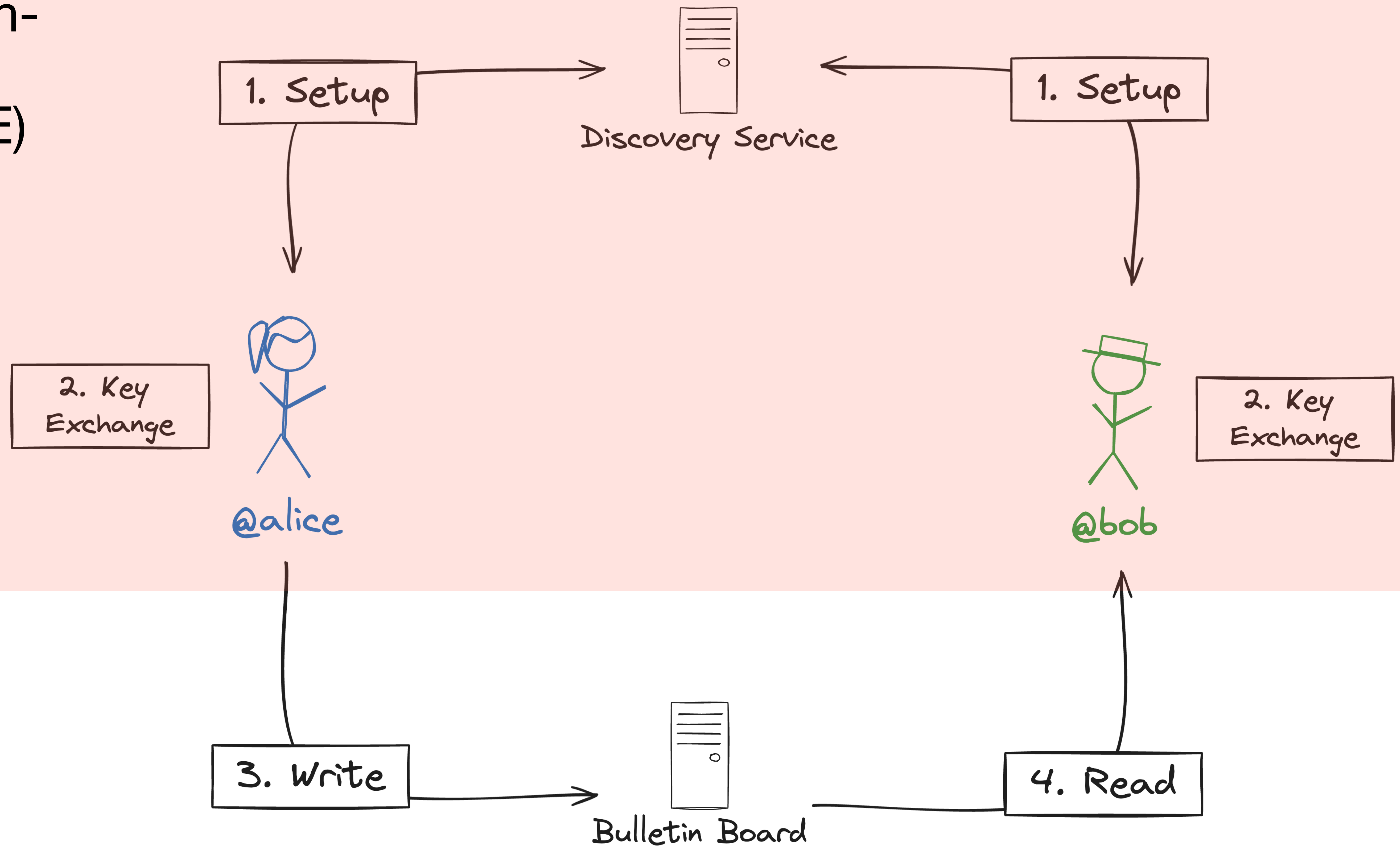
- Notice that:

$$\mathrm{Extract}(\mathrm{msk}, \mathrm{id}) = \boxed{H(\mathrm{id})^{\mathrm{msk}}}$$

- Does this look familiar?
- A private key is a **BLS signatures** on the identifier.
  - We can decentralise the TTP using **threshold BLS signatures**.
  - We can anonymise key requests by using **blind BLS signatures + ZKPoK** of the identifier (+ ownership).

Arke, System Overview

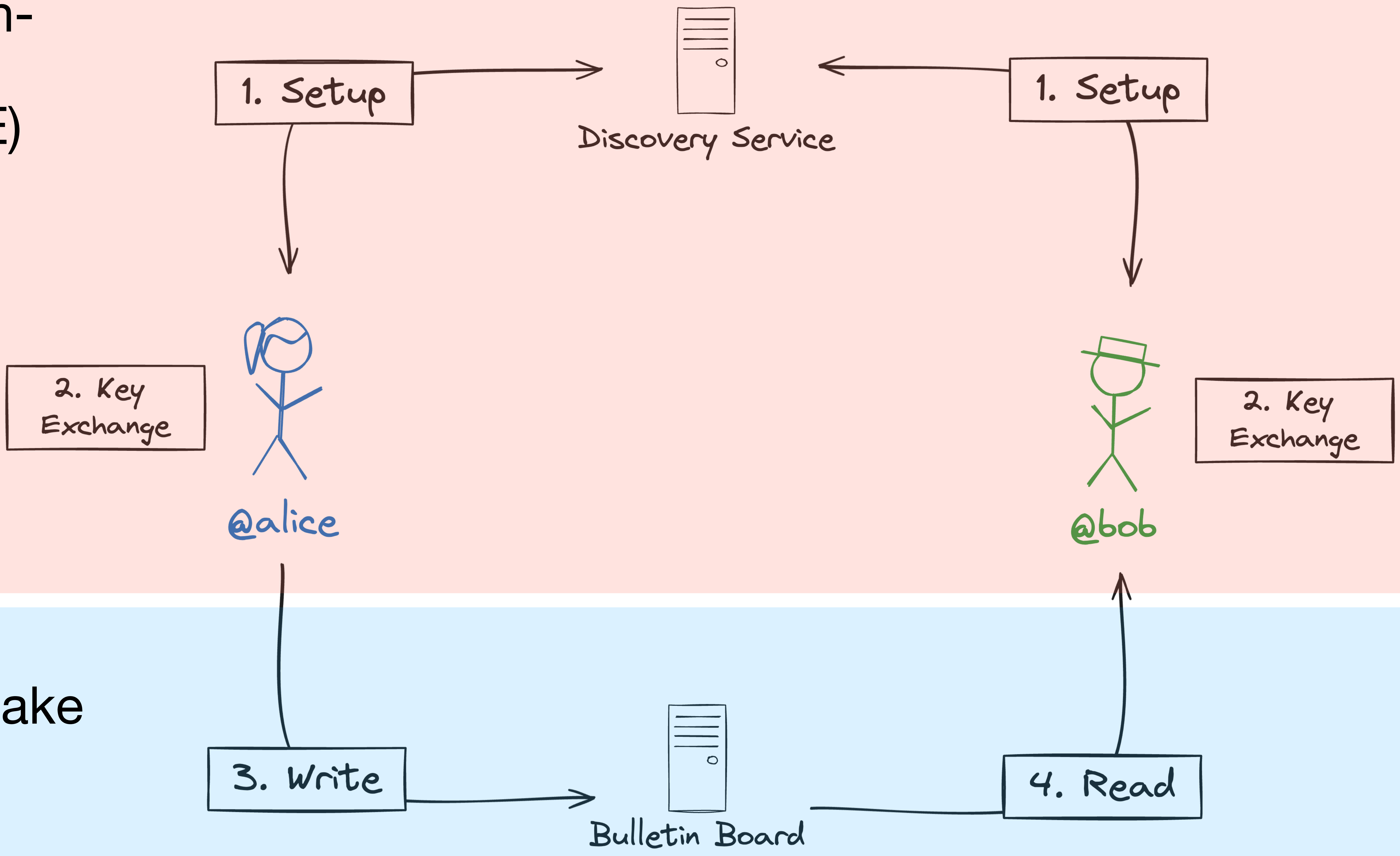
IDentity-based Non-Interactive Key Exchange (ID-NIKE)



Arke, System Overview

IDentity-based Non-Interactive Key Exchange (ID-NIKE)

Unlinkable Handshake



Unlinkable Handshake

Key	Value
tag_1	c_1
tag_2	c_2
…	…
tag_{n-1}	c_{n-1}
tag_n	c_n

- Given the shared secret $k_{\text{id},\text{id}'}$ compute:
    - an encryption key $k = \text{PRF}_0(k_{\text{id},\text{id}'})$.
    - a tag $t = \text{PRF}_1(k_{\text{id},\text{id}'})$.
    - a ciphertext $c = \text{Enc}_k(m)$.
- Publish (t, c) to the bulletin board.
- To read, derive k and t as above, then decrypt.
- Assumes that users can write to the BB via an **anonymity network** (e.g. Tor)

Arke

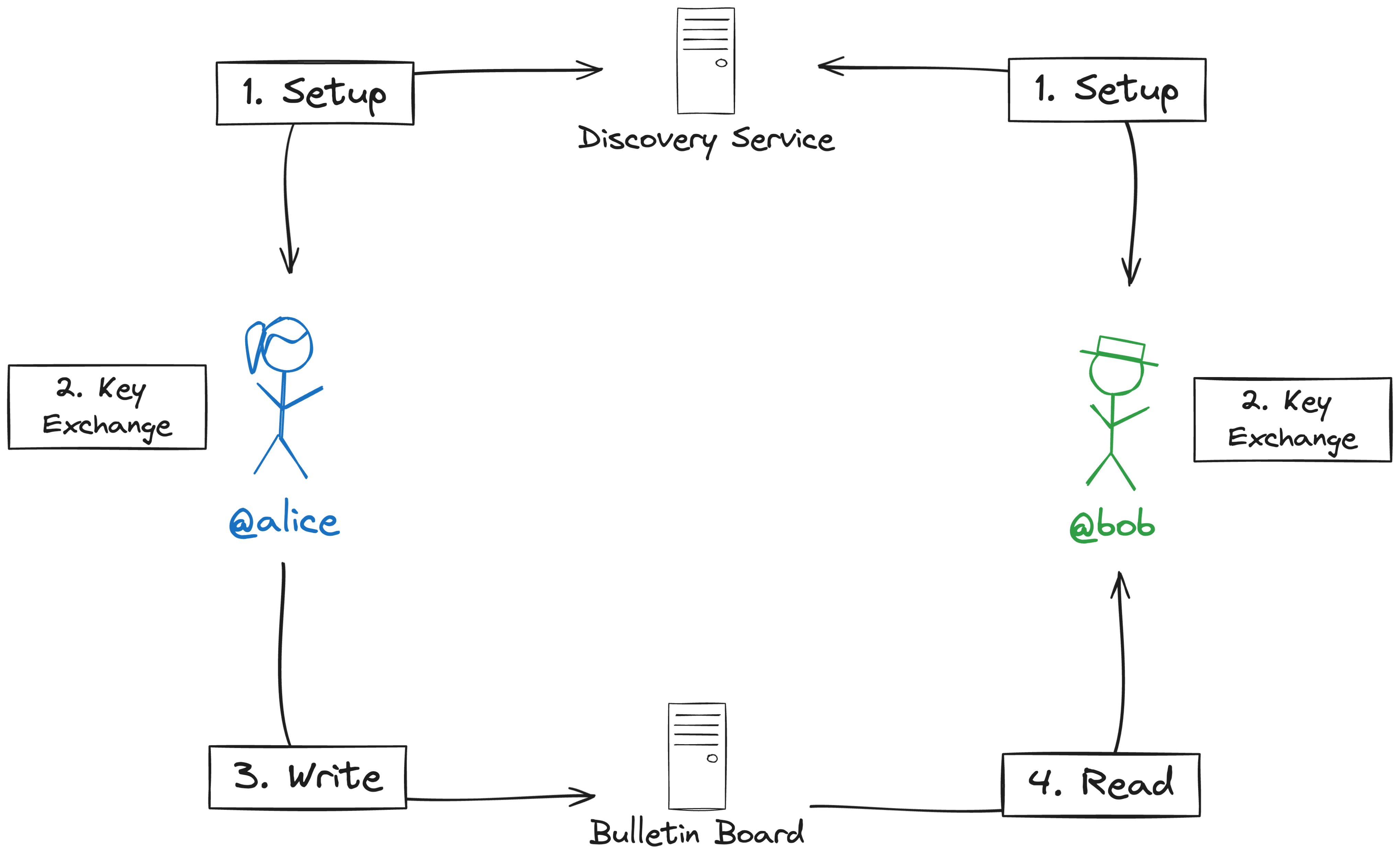
System Summary

1. Users anonymously obtain shares of their private key from the Discovery Nodes, and reconstruct their key locally.

2. Key exchange (1 pairing per friend).

3. Use the shared secret to derive encryption key, tag and ciphertext.

4. Use the shared secret to derive encryption key, tag and decrypt.



What else can we do?

- *Arke* allows to secretly exchange an **arbitrary message** with a known identity.
- Could send a pre-authorised transaction:
    - effectively paying a friend *before* they create an account - onboarding!
    - an airdrop that can be claimed even if the dropping party goes offline.
    - This construction is very similar to part of the Untraceable Transactions paper (UTT, ePrint 2022/452)
- Gaming: players from a same team can establish contact (?)
- Contact me if you have other ideas!

What you should not do!

- Use this system as a main communication channel.
- Send important, private information (e.g. private keys):
  - **ID-NIKEs do not provide forward secrecy**. If your key, your friend’s key or the master secret are compromised, your messages can all be read.
  - Better to send “public” info - e.g. addresses, public keys - or ephemeral secrets.

References

- Chaum, David, Mario Yaksetig, Alan T. Sherman, and Joeri de Ruiter. “UDM: Private User Discovery with Minimal Information Disclosure.” Cryptologia 46, no. 4 (2022): 347–79. doi:10.1080/01611194.2021.1911876.
- Clark, Mike. The Facts on News Reports About Facebook Data. April 2021 https://about.fb.com/news/2021/04/facts-on-news-reports-about-facebook-data/
- Hagen C, C. Weinert, C. Sendner, A. Dmitrienko, and T. Schneider, “All the Numbers are US: Large-scale Abuse of Contact Discovery in Mobile Messengers,” Cryptology ePrint Archive, Paper 2020/1119, 2020. [Online]. Available: https://eprint.iacr.org/2020/1119
- J.-H. Hoepman, “Privately (and unlinkably) exchanging messages using a public bulletin board,” in Proceedings of the 14th ACM Workshop on Privacy in the Electronic Society, 2015, pp. 85–94.
- Kales D, C. Rechberger, T. Schneider, M. Senker, and C. Weinert, “Mobile Private Contact Discovery at Scale,” in Proceedings of the 28th USENIX Security Symposium, 2019.
- Kelly M.J., You’ve been scraped, the Facebook data leak explained (April 2021). Online https://blog.mozilla.org/en/privacy-security/facebook-data-leak-explained/
- Kiss A,J.Liu,T.Schneider,N.Asokan,andB.Pinkas,“Private set intersection for unequal set sizes with mobile applications,” Cryptology ePrint Archive, 2017.
- Kocaoğullar, C., Hugenroth, D., Kleppmann, M., & Beresford, A. R. (2023). Pudding: Private User Discovery in Anonymity Networks.
- Marlinspike, Moxie. The Difficulty of Private Contact Discovery (2014). Online.
- Marlinspike, Moxie. Technology preview: Private contact discovery for Signal (2017). Online
- Mohnblatt, N., Sonnino, A., Gurkan, K., & Jovanovic, P. (2023). Arke: Scalable and Byzantine Fault Tolerant Privacy-Preserving Contact Discovery. Cryptology ePrint Archive, Paper 2023/1218. https://eprint.iacr.org/2023/1218
- Sakai R, K. Ohgishi, and M. Kasahara, “Cryptosystems based on pairing,” in The 2000 Symposium on Cryptography and Information Security, 2000, pp. 26–28.
- Tomescu, A., Bhat, A., Applebaum, B., Abraham, I., Gueta, G., Pinkas, B., & Yanai, A. (2022). UTT: Decentralized Ecash with Accountable Privacy. Cryptology ePrint Archive, Paper 2022/452. https://eprint.iacr.org/2022/452

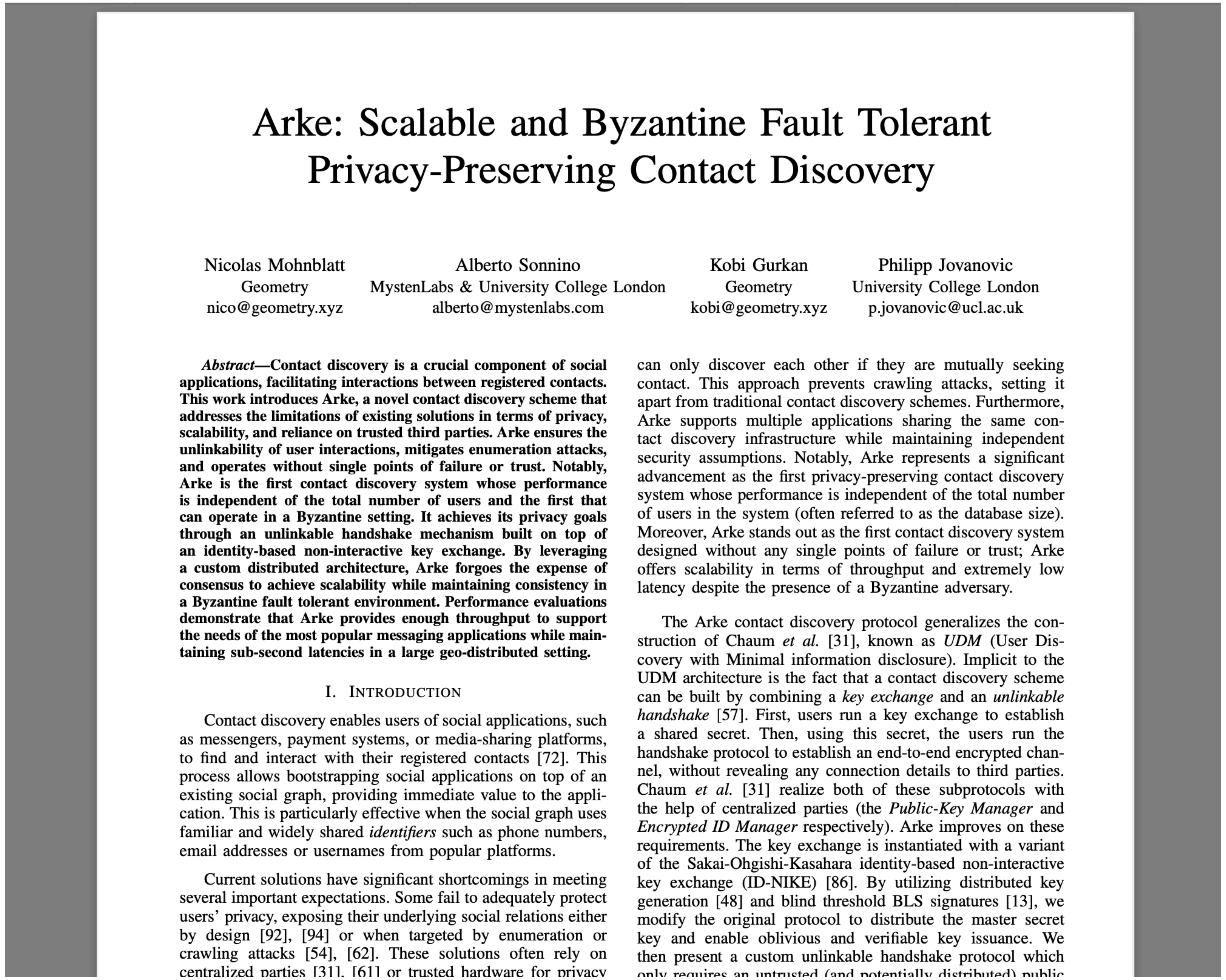
ePrint 2023/1218

PoC implementation:
https://github.com/asonnino/arke

Arke

Concluding remarks

- Today:
  - Contact Discovery
  - Related Work
  - ID-NIKE + Unlinkable Handshake
- Paper:
  - how to efficiently implement the bulletin board (without full consensus)
  - access control, spam prevention, storage cleanup, proofs of ownership and other **practical considerations**
  - **security proofs**
  - **experiments**: implement and evaluate the system in a geo-distributed setting



Arke: Scalable and Byzantine Fault Tolerant Privacy-Preserving Contact Discovery

Nicolas Mohnblatt
Geometry
nico@geometry.xyz

Alberto Sonnino
MystenLabs & University College London
alberto@mystenlabs.com

Kobi Gurkan
Geometry
kobi@geometry.xyz

Philipp Jovanovic
University College London
p.jovanovic@ucl.ac.uk

***Abstract*—Contact discovery is a crucial component of social applications, facilitating interactions between registered contacts. This work introduces Arke, a novel contact discovery scheme that addresses the limitations of existing solutions in terms of privacy, scalability, and reliance on trusted third parties. Arke ensures the unlinkability of user interactions, mitigates enumeration attacks, and operates without single points of failure or trust. Notably, Arke is the first contact discovery system whose performance is independent of the total number of users and the first that can operate in a Byzantine setting. It achieves its privacy goals through an unlinkable handshake mechanism built on top of an identity-based non-interactive key exchange. By leveraging a custom distributed architecture, Arke forgoes the expense of consensus to achieve scalability while maintaining consistency in a Byzantine fault tolerant environment. Performance evaluations demonstrate that Arke provides enough throughput to support the needs of the most popular messaging applications while maintaining sub-second latencies in a large geo-distributed setting.**

I. Introduction

Contact discovery enables users of social applications, such as messengers, payment systems, or media-sharing platforms, to find and interact with their registered contacts [72]. This process allows bootstrapping social applications on top of an existing social graph, providing immediate value to the application. This is particularly effective when the social graph uses familiar and widely shared *identifiers* such as phone numbers, email addresses or usernames from popular platforms.

Current solutions have significant shortcomings in meeting several important expectations. Some fail to adequately protect users' privacy, exposing their underlying social relations either by design [92], [94] or when targeted by enumeration or crawling attacks [54], [62]. These solutions often rely on centralized parties [31], [61] or trusted hardware for privacy

can only discover each other if they are mutually seeking contact. This approach prevents crawling attacks, setting it apart from traditional contact discovery schemes. Furthermore, Arke supports multiple applications sharing the same contact discovery infrastructure while maintaining independent security assumptions. Notably, Arke represents a significant advancement as the first privacy-preserving contact discovery system whose performance is independent of the total number of users in the system (often referred to as the database size). Moreover, Arke stands out as the first contact discovery system designed without any single points of failure or trust; Arke offers scalability in terms of throughput and extremely low latency despite the presence of a Byzantine adversary.

The Arke contact discovery protocol generalizes the construction of Chaum *et al.* [31], known as *UDM* (User Discovery with Minimal information disclosure). Implicit to the UDM architecture is the fact that a contact discovery scheme can be built by combining a *key exchange* and an *unlinkable handshake* [57]. First, users run a key exchange to establish a shared secret. Then, using this secret, the users run the handshake protocol to establish an end-to-end encrypted channel, without revealing any connection details to third parties. Chaum *et al.* [31] realize both of these subprotocols with the help of centralized parties (the *Public-Key Manager* and *Encrypted ID Manager* respectively). Arke improves on these requirements. The key exchange is instantiated with a variant of the Sakai-Ohgishi-Kasahara identity-based non-interactive key exchange (ID-NIKE) [86]. By utilizing distributed key generation [48] and blind threshold BLS signatures [13], we modify the original protocol to distribute the master secret key and enable oblivious and verifiable key issuance. We then present a custom unlinkable handshake protocol which only requires an untrusted (and potentially distributed) public