FinTracer with side information

Michael Brand

November 23, 2023

Executive Summary

We show how partner agency data can be utilised within the FinTracer framework.

1 Introduction

The FinTracer algorithm is a powerful and versatile algorithm that utilises the fact that in financial transactions both the ordering and the recieving reporting entity (RE) involved in a transaction are aware of its details. If we consider the financial system as a transaction graph, FinTracer utilises the fact that the edges of this graph are visible to both involved REs to allow searching and matching interesting typologies across the entire graph.

The problem, however, is that this assumption does not always hold. AUSTRAC works with partner agencies (PAs), and these PAs may provide relevant data regarding both people and relationships between people for specific intel investigations - such data is not part of the FinTracer graph and is not "two way viewable".

In this document, we deal at the algorithmic level with the question of how PA data can be utilised.

We note that PA data can range from unclassified, to high level security classifications. This document does not address the issue of using security classified data in any way. Other documents, particularly [2] describe the security model of the Purgles Platform.

This document describes new types of functionality that the FinTracer system may enable, in a situation where peer nodes include both REs and PAs.

Specifically, we describe the following.

1. How PAs can convert their person-centric data to account-centric information (Section 2), and
2. How PAs can incorporate their private information into FinTracer runs (Section 3).

2 Converting person-centric data

2.1 Background and motivation

In the present design of the FinTracer system, information is account-centric, and the system supports a wide range of account-centric questions that can move intel investigations forward. Where investigations require person-centric information, the assumption is that this information can be asked of the REs, and can be loaded manually by the REs into the system as required.

A typical case in point is one where AUSTRAC has, either in its own databases or in PA data, information about the identity of a person of interest (POI). AUSTRAC can ask the REs to generate a FinTracer tag indicating which accounts are held by this person, and this tag can then be used as either defining FinTracer source accounts or FinTracer destination accounts, as is needed by each specific query.

Such a methodology is, however, not feasible in the scenario discussed in the present document. Namely, we are interested in situations in which a PA has identity information about a POI but

does not know a complete list of accounts the individual owns - and the PA is unable, for example due to the information's classification, to share the POI's identity with REs. In this particular case, the present plan for the FinTracer system does not allow any method for either AUSTRAC or the PA to bridge the gap between person-centric information and account details, because no information linking the two is accessible to FinTracer.

In the discussion below, we must assume that we are considering a "future" version of the FinTracer system, in which account-holder data is automatically-available to the system. Specifically, we assume that the FinTracer Auxiliary Database includes a table, accessible to the FTIL user, linking account numbers with account holder information.

The availability of such a table would unlock much additional functionality of the FinTracer system. An example of which is privacy preserving record linkage - with the aim of linking together all accounts held by the same individual. In this section, we discuss how such a table, combined with FinTracer DARK functionality, enables PAs to securely link suspect information with account numbers.

Note 2.1. Notably, if all the PA wishes to do is to define the accounts belonging to a known set of suspects as FinTracer source accounts, no per-person linkage is really needed: it is enough to identify accounts belonging to "anyone" in the set of suspects. The linking algorithms we discuss here are meant for more complicated usages that do require such per-person linking, such as when trying to answer specific "Who" questions.

2.2 Brute force linkage

The first part of the linking algorithm is the generation of signatures. Members of the team have previously developed the pSig algorithm for entity resolution. pSig takes into account identify information including name, address, ID numbers and more. Each field is matched in complicated ways, leading to a fuzzy matching score, which, if exceeding a threshold, is taken as a match. Such fuzzy matching methods can be translated to a signature-based matching protocol by considering every possible minimal set of attribute matches that can lead the fuzzy algorithm to declare a match between two entities, and translating that to a signature that encapsulates this attribute subset, e.g. by means of a hash.

The exact algorithm for doing so was detailed in [1], and the total number of signatures that the algorithm should associate with any given person was bounded there at $\gamma < 124$.

At this point, one can take one of two approaches.

In the first approach, no matching between individuals and account numbers is ever explicitly done. Instead, communication between PAs and REs is done entirely in terms of DARK signatures, using the Extreme DARK algorithm [7]: each person and each account is described by their up-to-γ signatures, and the Extreme DARK algorithm propagates messages between the two.

To recap the Extreme DARK algorithm: from a user's perspective, all DARK algorithms allow one to do the following.

1. Each peer node can define a set of named "mailboxes".
2. Each peer node can send encrypted tag values to any mailbox.

This is done while preserving the following privacy guarantees.

- AUSTRAC does not gain private information regarding what the sending account was, what the recipient account was, what the mailbox identity signifies or what the sent tag value was.
- The sending party does not gain information regarding the receiving accounts: it cannot tell to whom and even whether any of its encrypted values were delivered; as far as it can tell, the values may have been delivered to no account or to many accounts at each of the other peer nodes.

- The receiving party does not gain information regarding the sending accounts: it cannot tell from whom and even whether any encrypted tags were delivered to it; as far as it can tell, the values may have been delivered from no account (in which case, they are zero values imputed by AUSTRAC) or from many accounts at each of the other peer nodes.

- Neither the sending nor the receiving party gains any information regarding the delivered message itself. This is an encrypted tag value that can only be read by AUSTRAC.

The Extreme DARK algorithm is distinguished from other DARK algorithms only in the extent to which it is resilient in delivering these privacy guarantees.

In this particular case, we use the at-most-γ minimal signatures as DARK mailbox addresses, and in this way propagate the tag values. This is a mechanism that allows propagating tags not just between REs, but between any peer nodes, with the idea that in this case it can be used to propagate tags between PAs (using person-centric tags, where each value corresponds to a person) and REs (using account-centric tags, where each value corresponds to an account).

2.3 A more advanced linking algorithm

The above is a viable solution approach, but it means that any PA query must involve message passing among γ-times-the-number-of-RE-customers mailboxes—a very heavy FinTracer DARK run, and this on top of the Extreme DARK propagation algorithm that is very heavy to begin with.

The alternative is to perform some preliminary work, which speeds up later querying.

The main idea here is to provide AUSTRAC with enough information for it to be able to function as a "postmaster" of sorts (which is, essentially, its role in every DARK algorithm). AUSTRAC will not gain any information regarding the identity of any persons of interest. Rather, these will be anonymised to AUSTRAC as "Individual #1", "Individual #2", and so on.

AUSTRAC will also not gain any information regarding RE accounts. Rather, these will be anonymised to AUSTRAC as "Account #1", "Account #2", and so on.

What AUSTRAC does discover in this algorithm is the matching under this anonymisation. In other words, it will discover that, e.g., "Individual #1 matches Account #5", but only the PA will know the identity of "Individual #1" and only the RE will know the identity of "Account #5". (Meanwhile, it is only AUSTRAC that discovers the matching. Neither the RE nor the PA are privy to it.)

One can only use the more advanced linking methods (and reap the associated saving in computational costs) under the assumption that this level of discovery by AUSTRAC, relevant only to individuals of interest but to all their accounts, is acceptable. If it is not acceptable, one can revert to the brute-force algorithm described in Section 2.2, and all functionality described in this document can be attained by straightforward (but quite heavy) usage of Extreme DARK.

Assuming that this level of discovery for AUSTRAC is acceptable, the anonymised matching is the minimal amount of information required for AUSTRAC.

To show how this information can be revealed to AUSTRAC, suppose that there are n individuals, $p_1, \ldots, p_n$, that need matching, and that the PA has a person-centric FinTracer tag, x, associated with them. In other words, we are taking $x[\tilde{p}_1], \ldots, x[\tilde{p}_n]$ to be the tag values associated with $p_1, \ldots, p_n$, respectively, although the actual relationship between p_i and $\tilde{p}_i$ is arbitrary, and should, for optimal anonymisation, be random.

On the RE end, let $\{t_i\}_{i=1}^n$ be normal, account-centric FinTracer tags, except that instead of associating each $t_i[\tilde{a}]$ with the corresponding account, a, the RE uses a random permutation, π, known only to itself, so that the tag value associated with a is actually $t_i[\pi(\tilde{a})]$. This permutation is what provides AUSTRAC with anonymisation for the account identities.

Algorithm 1 uses these in a way akin to a previously described pair matching algorithm.[1]

[1] For pair finding, this algorithm was since superseded by [3], but for our present purposes the simple matching algorithm is the appropriate one to use.

Algorithm 1 Matching PA persons of interest with RE accounts

1: **for** $i \in 1, \ldots, n$ **do**
2: Set $x[\tilde{p}_i]$ to 1 and all other $x[\tilde{p}_j]$ to 0.
3: Use Extreme DARK to propagate the values from x to t_i, using the signatures of the identities of the persons of interest as the PA's mailbox addresses and the signatures of the identities of all account owners as the RE mailboxes.
4: Use the oblivious sparse querying algorithm of [8] for AUSTRAC to retrieve the set $\{\pi(a_1^i), \ldots, \pi(a_{m_i}^i)\}$ of all accounts associated with p_i.
5: **end for**

3 Using PA information in FinTracer runs

Now that a link has been created between the PAs' person-centric data and the REs' account-centric data, let us investigate the various ways in which PAs can inject their data into a FinTracer computation.

The idea is to enable as-usual FinTracer runs, with all of FinTracer's usual capabilities in defining typologies, but to incorporate

1. PA information about people,
2. PA information about relationships between people, and
3. Computation results that are visible only to PAs, and that AUSTRAC itself is not privy to.

3.1 Tag propagation

To show how all this is done, consider for a moment how standard FinTracer tag propagation works in this scenario.

First, REs and PAs can both set accounts/people as "source" nodes in the FinTracer computation by associating them with nonzero tag values. All other tag values are set to zero. Much like REs incorporate their knowledge about accounts when choosing which to set to a nonzero tag value at any time, so do PAs incorporate their information about people when choosing the tag value associated with any person-of-interest at any point.

Once tags have been set, they need to be propagated. The standard FinTracer algorithm allows us to propagate tags from one RE to another, as well as between different accounts managed by the same RE.

To propagate tags from PAs to REs, we use a combination of Extreme DARK and the oblivious tag-setting algorithm of [4]. The full protocol is as follows, for propagating a tag value from person p_i to the accounts associated with p_i at RE r. We assume, throughout, that n DARK mailboxes $m_1, \ldots, m_n$ have been defined, one for each person of interest.

1. The PA refreshes the tag.
2. The PA encrypts the tag value v using the public key of r. (The value of v is now double-encrypted: once with the public key of AUSTRAC and once with the public key of r.)
3. The PA places the encrypted tag value in m_i.
4. AUSTRAC uses the algorithm of [4] to obliviously set all tag values associated with accounts of p_i to the encrypted value of v.
5. The RE undoes the encryption to restore the value of v, leaving only the encryption layer using AUSTRAC's public key.
6. The RE undoes the permutation π to move the tag values to their appropriate final destination.

To propagate tag values from the REs to the PA, the exact same procedure is applied, only in the opposite order, and using the oblivious tag reading algorithm of [5] instead of the oblivious tag setting algorithm of [4].

Finally, we note that it is just as easy for a PA to propagate tag values to itself as it is for an RE to propagate tag values between accounts that it manages. In this way, the PA can incorporate privately into the computation known relationships between people. (For example, if two people, A and B, are known to be close relatives, it is possible for a tag value to be propagated from the tag values associated with the accounts of A to $x[\tilde{A}]$ at the PA, to $x[\tilde{B}]$ at the PA, to the tag values associated with the accounts of B.)

Note 3.1. If more than one PA is involved in the computation, linkage between person-centric data between PAs can be performed as per Section 2. Once this is done, each person is associated with a mailbox, as before, and AUSTRAC can propagate messages among these mailboxes using standard Extreme DARK techniques.

3.2 Reading the results

The one remaining aspect in which using PA information is different to normal FinTracer runs is in the reading of the results. The reason for this is that in a standard run it is expected that the final results will be provided to AUSTRAC. Here, in contrast, the results need to ultimately reach a particular PA, and, in fact, it may be the case that AUSTRAC is not privy to the results at all.

Here, we identify two separate scenarios.

1. A situation where the final result is a tag value associated with an RE account, and
2. A situation where the final result is a tag value associated with a PA person of interest.

In both of these scenarios, it is possible to use the key-switching tools of [6] for the party holding the result to encrypt the tag value using a public key of the PA, send it to AUSTRAC for AUSTRAC to remove the encryption by its own key, and then have AUSTRAC send the result to the PA for final decryption.

With some limits, this works also for the more complicated querying algorithms that have been devised for FinTracer. For example, if the PA wants to receive the results of a sparse, oblivious query, as described in [8], the same key-switching process can be performed for each encrypted value sent from the RE. This will cause the PA to be able to read the result, but not AUSTRAC.

Alternatively, in the case in which the final result is a tag value associated with a PA person-of-interest it is possible for the PA to send to AUSTRAC tag values for decryption without a further encryption layer, only permuting the values with a random order permutation in order to prevent AUSTRAC from associating tag values with particular people (even under the anonymisation).

This alternative has the advantage that AUSTRAC can inspect the tag values and can decide not to forward these to the PA in case there are too many "match" results.

Note 3.2. In any case, it is important to note that in this scenario the PA is likely to play a role in all aspects of the run of the algorithm, from defining the initial query, to propagating the tags, to receiving the final results. This inevitably makes the PA a powerful player in this protocol, so a certain level of lack-of-protection from a PA who does not follow the protocol cannot be avoided.

An example of this is that the PA can send any tag values it wishes for AUSTRAC to decrypt in the final step of the algorithm. As such, any tag value that at any point passes through the PA can, in principle, be decrypted by the PA if it chooses to diverge from the algorithm's agreed protocol.

The present algorithm does not protect against such an eventuality, and in general we believe no algorithm can fully protect against such eventualities, either. Policies and procedures which determine and govern the appropriate use of the system are also a critical part of overall system security.

References

[1] Brand, M. (2020), Fast, Lightweight, Secure Record Set Intersection Without Homomorphic Encryption. (Available as: `fast_intersection.pdf`)

[2] Brand, M. (2022), *The Complete, Authorised and Definitive FinTracer Compendium.* (Available as: `FT_combined.pdf`)

[3] Brand, M. (2020), Efficient connection-finding with FinTracer. (Available as: `pair_finding.pdf`)

[4] Brand, M. (2020), Oblivious setting of source accounts in FinTracer. (Available as: `oblivious_a.pdf`)

[5] Brand, M. (2021), Oblivious querying of FinTracer tag values. (Available as: `oblivious_b.pdf`)

[6] Brand, M. (2021), ElGamal key switching. (Available as: `keyswitch.pdf`)

[7] Brand, M. (2021), Extreme DARK: A more secure DARK variant. (Available as: `extreme.pdf`)

[8] Brand, M. (2021), Oblivious (and non-oblivious) discovery of sparse matches. (Available as: `oblivious_c.pdf`)