Investigating the Privacy vs. Forwarding Accuracy Tradeoff in Opportunistic Interest-Casting

Gianpiero Costantino, Fabio Martinelli, Paolo Santi
IIT-CNR, Pisa, Italy
Email: name.surname@iit.cnr.it

Abstract—Many mobile social networking applications are based on a "friend proximity detection" step, according to which two mobile users try to jointly estimate whether they have friends in common, or share similar interests, etc. Performing "friend proximity detection" in a privacy-preserving way is fundamental to achieve widespread acceptance of mobile social networking applications. However, the need of privacy preservation is often at odds with application-level performance of the mobile social networking application, since only obfuscated information about the other user's profile is available for optimizing performance.
In this paper, we study for the first time the fundamental tradeoff between privacy preservation and application-level performance in mobile social networks. More specifically, we consider a mobile social networking application for opportunistic networks called interest-casting. In the interest-casting model, a user wants to deliver a piece of information to other users sharing similar interests ("friends"), possibly through multi-hop forwarding. In this paper, we propose a privacy-preserving friend proximity detection scheme based on a protocol for solving the Yao's "Millionaire's Problem", and we introduce three interest-casting protocols achieving different tradeoffs between privacy and accuracy of the information forwarding process. The privacy vs. accuracy tradeoff is analyzed both theoretically, and through simulations based on a real-world mobility trace. The results of our study demonstrate for the first time that privacy preservation is at odds with forwarding accuracy, and that the best tradeoff between these two conflicting goals should be identified based on the application-level requirements.

Index Terms—Opportunistic Networks, Interest-casting, Opportunistic Forwarding, Privacy, Secure Multiparty Computation.

1 Introduction

With the increasing penetration rate of smartphones, tablets, etc., mobile social networks are being considered the natural evolution of online social networks. Mobile social networks display several advantages over online social networks, such as larger potential user base – more than 5.6 billions mobile phone subscribers [1] vs. about 500 millions broadband Internet users [2], enabling location-aware social applications, possibility of ubiquitously running social networking applications without Internet access, and so on.

Mobile social networking applications have been recently introduced in the market [3], [4], [5], as well as in the academic community [6], [7], [8], [9], [10]. As mentioned above, mobile social networks enable novel, location-based services, such as friends proximity detection. In the context of mobile social networks, the term *friend* is used to refer to a person (potentially, a stranger) with whom a user might be interested in getting in touch with, where the notion of friendship used for detection depends on the specific application scenario.

While some mobile social network applications are based on centralized detection of friend proximity [3], [4], [6], [7], recent developments suggest using short-range wireless interfaces available on the portable device (typically, Bluetooth and/or WiFi) for fully-distributed friends proximity detection [9], [10]. Fully-distributed friends proximity detection approaches present potential advantages vs. centralized ones, such as possibility of operation in isolation from the Internet, lack of a single point of failure, scalability, etc.

Typically, a fully-distributed friend proximity detection approach operates as follows: $Phase\ 1)$ a user (Alice) periodically performs a neighbor discovery process to detect nearby devices; $Phase\ 2)$ once a new user's (Bob) device is detected, Alice's mobile social networking software automatically starts a friendship estimation procedure, which typically requires computing a similarity metric between Alice's and Bob's profiles; $Phase\ 3)$ if (and only) friendship estimation is successful, the software on Alice's and Bob's devices alerts them of the detected nearby friend, so that Alice and Bob can start talking, exchanging text messages, files, and so on.

Fully-distributed friend proximity detection raises serious privacy concerns: if users profiles are exchanged in plain text, Bob can acquire sensible information from Alice's profile (and vice-versa), such as her interests, home address, work place, political opinions, sexual orientation, etc. (the specific information leaked depending on how user profile is defined). If Bob is a malicious user, he can easily perform serious attacks such as identity cloning, selling Alice's personal information to third parties, and so on.

A number of solutions have been recently proposed to detect friend proximity in a privacy-preserving man-

ner. However, most of them are concerned with coarse-grained matching of users profiles, i.e., on detecting the number of common items (interest topics, names in the contact list, etc.) in the user profiles [11], [12]. Only recently, a few privacy-preserving approaches have been proposed to match fine-grained users profiles [13], [14] – e.g., profiles reporting a user's degree of interest in specific topics, including our work [15]. As observed in [14], fine-grained profile matching allows implementation of a wider class of mobile social networking applications than that enabled by coarse-grained profile matching.

Differently from previous works [11], [12], [13], [14] that only consider privacy-preserving profile matching, in this paper we investigate for the first time the interplay between *privacy preservation* and the resulting *application-level performance* of the mobile social networking application. To this purpose, we consider a specific mobile-social networking application, namely, the interest-casting application recently proposed in the context of opportunistic networks [16].

In interest-casting, each user in the network is characterized by a (fine-grained) profile expressing his/her degree of interest in different topics, and is both *provider* and *consumer* of information. A user is interested in exchanging information only with users sharing similar interests, where interest similarity is computed based on the user interest profiles. Notice that in the interest-casting application proposed in [16] information can propagate multi-hop among users, based on the store-carry-and-forward mechanism typical of opportunistic networks [17].

In this paper, we present, for the first time in the literature, a number of privacy-preserving versions of interest-casting. The need of privacy preservation in interest-casting stems from the observation that a user's interest profile can contain sensible information such as political opinion, sexual orientation, etc., and an individual, while in general keen on exchanging information with a stranger having, say, similar political opinions, might not want to disclose such a sensitive information to a stranger with different political views.

The different versions of interest-casting considered in the paper share the common feature of allowing tuning the level of privacy-preservation by means of a parameter used to compute the profile similarity metric. Thanks to this feature of the designed protocols, we are able to investigate the tradeoff between *privacy-preservation* and *accuracy of the forwarding process* in opportunistic interest-casting. In particular, we: i) define an information-theoretic metric to quantify privacy-preservation; ii) use the well-known *coverage* and *precision* metrics borrowed from information retrieval to quantify forwarding accuracy, i.e., the mobile social networking application ability to deliver information to all the interested users (coverage), and only to them (precision); and iii) investigate the inherent tradeoff between privacy-preservation and forwarding accuracy both analytically and through simulation.

Our specific technical advancements over the state-of-the-art are the following:

- the definition of a family of privacy-preserving forwarding protocols for interest-casting applications based on a protocol to solve the well-known Yao's "Millionaire's Problem" [18]. Each of these protocols allows tuning the privacy-preservation vs. forwarding accuracy tradeoff by means of a parameter used in the computation of interest similarity;
- the definition of an information-theoretic notion of privacy-preservation, aimed at estimating the amount of Bob's uncertainty about the content of Alice's interest profile *before* and *after* the execution of the profile matching protocol. Differently from existing works that characterize privacy preservation in terms of a *property* the protocol must fulfill [11], [13], [14], the one proposed in this paper is a *metric* taking continuous values in the $[0, 1]$ interval, with 0 and 1 expressing *complete privacy leakage* and *full privacy preservation*, respectively. As we shall see, the definition of a *continuous* privacy-preservation metric is the pre-requisite for formally investigating the privacy-preservation vs. forwarding accuracy tradeoff;
- the investigation of the interplay between privacy preservation and forwarding accuracy in a *complete* mobile social networking application, namely, interest-casting. While attention in existing work [11], [13], [14] is focused on the profile matching phase only of a mobile social networking application – $Phase\ 2$) of a typical application as previously described –, in this work we mostly focus the attention on how privacy preservation requirements impact the performance of a mobile social networking application, i.e., we also analyze the $Phase\ 3$) of a typical mobile social networking application, which is the most important one since it is the phase enabling the real social interaction between users.

The rest of this paper is organized as follows. In Section 2 we discuss related work. Section 3 introduces the network model and the interest-casting application. In Section 4, we define the specific forwarding protocols considered in our work, and study some of their properties. Section 5 discusses the challenges related with privacy-preserving interest-casting, while Section 6 introduces an optimised version of The Millionaire's Problem to realize privacy-preserving forwarding. In Section 7, we theoretically analyze the privacy-preservation vs. forwarding accuracy tradeoff, while we analyze the same tradeoff by means of simulations in Section 8. Finally, Section 9 concludes the paper.

2 Related Work

2.1 Mobile social networking applications

In this section, we present recently proposed *fully-distributed* mobile social networking applications, that are more relevant to our work.

Nokia Sensor [5] is an application available on some Nokia cell phones that uses the Bluetooth (BT) interface to discover nearby users running the same application; once a new user is discovered, her/his profile (called *folio*) can be visualized, and a connection possibly established for free messaging, file exchange, etc.

PeopleNet [8] is a system for multi-casting information and querying a group of devices connected by a mobile ad hoc network established through BT/WiFi links.

In [9], the authors present an extension of the well-known Twitter social network to opportunistic networks. The basic idea is that users with a Twitter account can opportunistically receive and send "tweets" also through the BT interface, and then these "tweets" are propagated in the opportunistic network in an epidemic fashion.

In [10], the authors present a mobile social networking application aimed at assisting users in starting a small talk with nearby users sharing similar (coarse-grained) interests. Similarly to the above approaches, also the E-SmallTalker application of [10] uses the BT interface to discover nearby users and perform profile matching.

Other examples of social networking applications for opportunistic networks are [19], [20], [21].

All mobile social networking applications mentioned above are not concerned with privacy preservation: when profile matching is needed to determine whether to establish a connection, profiles are exchanged in plain text, opening the way to potentially serious security attacks as described in the Introduction.

2.2 Privacy-preserving profile matching

Following [14], profile matching protocols can be divided into *coarse-grained* and *fine-grained* approaches. In the former approach, the user profile is defined in terms of a set of items taken from a common universe $\mathcal{I}$, where items can represent attributes, interest topics, names in a contact list, etc. The profile is coarse-grained in the sense that, for each item $i \in \mathcal{I}$, we have that either i belongs or it does not belong to a specific user's profile. For instance, if items represent interest topics, a user might indicate "cinema" in her/his interest profile, but she/he has no way of expressing the degree of interest in the topic. Conversely, fine-grained profile matching protocols allows a user to express different degrees of interest in each item of the universe. Typically, a user's interest in a certain topic i is expressed by means of an integer taking value in an interval $[0, max]$, where 0 and max denote no and maximal interest in a topic, respectively. Since fine-grained profiles are a generalization of coarse-grained ones, it is clear that fine-grained profile matching protocols are more general and powerful than coarse-grained ones, and can be used to implement a larger class of mobile social networking applications.

Privacy-preserving profile matching protocols are typically based on security protocols designed within the realm of secure multi-party computation [22], and more specifically secure two-party computation. In secure two-party computation, the problem is to allow two parties to jointly compute the outcome of a function $f(x_1, x_2)$ whose input values x_1, x_2 are held by the single parties, without revealing more than the information provided by the output itself to the other party. Protocols for coarse-grained profile matching are typically based on privacy-preserving set intersection computation [11], [12], while existing fine-grained profile matching protocols are based on privacy preserving vector dot product [13] and ℓ_1 norm [14] computation. Independently of the specific function used to estimate profile matching, all the above mentioned approaches share the property that, when privacy-preservation of the protocol at hand is investigated, this is defined in terms of a certain *privacy requirement* the protocol must fulfill. For instance, it might be required that, at the end of the protocol execution, both parties involved in the computation know the function f which is jointly computed, and the outcome of the joint computation [13], [14]. A stronger privacy requirement might be that only a party involved in the computation knows the function f used to estimate profile matching [13]. Furthermore, all existing approaches are concerned only with the design of privacy-preserving profile matching, and do not consider the impact of, say, different privacy requirements on the mobile social networking applications running on top of profile matching.

Our work differs from existing approaches under the following respects:

- we define an information-theoretic notion of privacy-preservation expressed as a continuous value in the $[0, 1]$ interval. Thus, we go beyond the notion of privacy requirement, and we put forward a more general privacy-preservation metric that can be used to *quantitatively* estimate privacy preservation, also across different protocols;
- we study the impact of using stricter/looser privacy-preservation requirements during the profile matching phase on the application-layer performance of a specific mobile social networking application; namely, interest-casting.

3 Network Model and Preliminaries

We consider an opportunistic network (OppNet) composed of n nodes (users), and denote the set of nodes in the network by $\mathcal{N}$. Similarly to [16], we assume user interests can be modelled as an m-dimensional vector in a common m-dimensional *interest space*, where $m \ll n$. More formally, the *interest profile* of user A is defined as:

$$I_A = (a_1, \ldots, a_m) \ ,$$

where $a_i \in [1, max]$ is an integer representing A's interest in the i-th topic of the interest space. Note that interests are expressed as integers in the range $[1, max]$, with 1 representing no interest and max (an arbitrary integer > 0) representing maximum interest[1]. Although our ap-

1. The notion of interest profile can be straightforwardly extended to represent also information about a user's habits, such as living in a certain neighborhood, working in a certain place, and so on. For details, see [16].

proach can be extended to deal with the case of two users with the same interest profile, to simplify presentation in the following we make the assumption that no two users in the network have the same interest profile.

In this paper, we are concerned with realizing a privacy preserving *interest-casting* primitive, where the interest-casting primitive is defined as follows [16]. Let S be a user denoted as the message *source*. The message M generated by S must be delivered to all nodes in the set $\mathcal{D}(S,\gamma)$, where

$$\mathcal{D}(S,\gamma) = \{U \in \mathcal{N} - \{S\} | sim(U,S) \geq \gamma\} \ ,$$

where $sim(U,S)$ is a similarity metric used to express similarity between U and S's interest profiles, with relatively higher similarity values representing relatively more similar interests, and γ is the *relevance threshold*. Set $\mathcal{D}(S,\gamma)$ is called the set of *relevant destinations*, and in principle it is not known in advance to node S. Instead, set $\mathcal{D}(S,\gamma)$ is implicitly defined by S's interest profile, and by the relevance threshold γ. Furthermore, users in set $\mathcal{D}(S,\gamma)$ are not assumed to undertake any explicit action (e.g., subscribing to a thematic channel) to be able to receive message M. This is in sharp contrast to more traditional networking primitives such as multicast, where the set of destinations is known in advance to the source, and publish/subscribe, where subscriptions to thematic channels are mandatory.

More specifically, in this paper we define the following similarity metric between interest profiles, which we call *vector-component-wise* (vcw) similarity metric[2]. Let $S = (s_1,\ldots,s_m)$ and $U = (u_1,\ldots,u_m)$ be the interest profiles of users S and U, respectively. We have:

$$vcw(U,S,\lambda) = \begin{cases} 1 & \text{if } \forall i \in \{1,\ldots,m\},\ |u_i - s_i| \leq \lambda \\ 0 & \text{otherwise} \end{cases} \ ,$$

where $\lambda \in [0, max]$ is an integer parameter used to narrow/widen the scope of the interest-cast[3]. More specifically, by setting $\gamma = 1$, we have that $\mathcal{D}(S,1) = \mathcal{N}$ if $\lambda = max$, and $\mathcal{D}(S,1) = \emptyset$ if $\lambda = 0$. To simplify notation, in the following we denote $\mathcal{D}(S,1)$ by $\mathcal{D}(S)$.

We assume message M generated by S is characterized by a TimeToLive (TTL), i.e., a time interval beyond which the information contained in the message is considered no longer valuable. The goal of the forwarding protocols described in the following is delivering a copy of M to as many nodes in $\mathcal{D}(S)$ as possible within time TTL since its generation at S. More specifically, for a given forwarding protocol $\mathbf{F}$, and denoting by $\mathbb{P}_{\mathbf{F}}(U)$ the property "user U received a copy of M within time TTL under forwarding scheme $\mathbf{F}$", we define the set of *covered nodes* $\mathcal{C}(\mathbf{F})$ as follows:

$$\mathcal{C}(\mathbf{F}) = \{U \in \mathcal{N} | \mathbb{P}_{\mathbf{F}}(U) \text{ is } \mathbf{true}\} \ .$$

We can now define the following *precision* and *coverage* metric (equivalent to the precision and recall metrics well known in information retrieval [23]). We have:

$$Prec(\mathbf{F}) = \frac{|\mathcal{C}(\mathbf{F}) \cap \mathcal{D}(S)|}{|\mathcal{C}(\mathbf{F})|}$$

and

$$Cov(\mathbf{F}) = \frac{|\mathcal{C}(\mathbf{F}) \cap \mathcal{D}(S)|}{|\mathcal{D}(S)|} \ ,$$

where $Prec(\mathbf{F}) = 1$ represents maximum possible precision (M is delivered only to nodes in $\mathcal{D}(S)$), and $Cov(\mathbf{F}) = 1$ represents maximum possible coverage (M is delivered to all nodes in $\mathcal{D}(S)$). Ideally, we would like to design a forwarding protocol simultaneously achieving maximum precision and coverage. However, as we shall see in the following, the two metrics above are often in contrast with each other, and the most adequate tradeoff between them should be sought. The notation used in this paper is summarized in Table 1.

TABLE 1
Notation table

Symbol	Explanation
S	node generating the message
U, V	generic nodes
M	generated message
m	number of topics in the interest profile
max	maximum interest value for a topic
λ	threshold for computing the similarity metric
k	number of tested topics for similarity computation

4 Forwarding Protocols

In the following, we will present privacy-preserving versions of the following forwarding protocols:

- *direct delivery* (**DD**): strictly speaking, this is not a forwarding protocol: source node S delivers a copy of M whenever it has a communication opportunity with a node $U \in \mathcal{D}(S)$. Message forwarding is not allowed: only S can deliver copies of M to relevant destinations.
- *2-hop forwarding* (**2H**): similarly to **DD**, node S delivers a copy of M to each node in $\mathcal{D}(S)$ it gets in touch with. However, in this case forwarding of a copy of M to other nodes is allowed. More specifically, any node U in $\mathcal{D}(S)$ holding a copy of M can deliver a copy of it to any other node V it meets under the condition that $vcw(U,V,\lambda) = 1$. Note that, in order to preserve a minimum level of precision, forwarding can occur only along paths composed of two hops at most: in particular, any node which receives a copy of M from a node $U \neq S$ (as node V above) is not allowed to further forward the message.
- *restricted 2-hop forwarding* (**R2**): similarly to **DD** and **2H**, node S delivers a copy of M to each node in $\mathcal{D}(S)$ it gets in touch with. Similarly to **2H**, two-hops forwarding of a copy of M is allowed, however under a stricter condition than in case of **2H**. In

2. In the original definition of interest-cast [16], the authors used the cosine metric for similarity. Unfortunately, using cosine metric in the context of secure two-party computation is highly non trivial. For this reason, we use instead the vcw similarity metric, which is equivalent to one of the ℓ_1-norm similarity metrics defined in [14].

3. Notice that, while in principle it is possible to define vcw similarity using different thresholds in each topic, using a single threshold λ for all topics is preferable to simplify notation, as well as to ease the definition of the privacy preserving metric, and the simulation experiments.

fact, a node U which received message M from S is allowed to act as forwarder if and only if $vcw(U, S, \lambda') = 1$, where $\lambda' < \lambda$. When a forwarder node U meets another node V, it can deliver a copy of M to V under the condition that $vcw(U, V, \lambda') = 1$. As we shall see in the following, this restrictive forwarding rule allows, by suitably tuning parameter λ', to optimally address the precision/coverage tradeoff.

4.1 Properties of forwarding protocols

In this section, we state some properties of the forwarding protocols considered in this paper. First, we observe that protocol **DD** always achieves maximum precision, i.e., $Prec(\mathbf{DD}) = 1$. In fact, according to protocol **DD**, only nodes in set $\mathcal{D}(S)$ can receive a copy of M. Notice, however, that this protocol likely displays low coverage, since no forwarding mechanism is realized; i.e., relatively few communication opportunities can be exploited to deliver M to relevant destinations.

Protocol **2H** aims at increasing coverage introducing two-hops forwarding. However, this comes at the price of precision. In fact, the following example shows that under protocol **2H**, also nodes in $\mathcal{N} - \mathcal{D}(S)$ can receive M. Assume $m = 1$, $max = 100$, $\lambda = 10$, and the following three nodes with respective interest profiles are part of the network: $S = (50)$, $U = (59)$, and $V = (65)$. Consider a message generated at node S. Given the parameter setting as above, we have $U \in \mathcal{D}(S)$ and $V \in \mathcal{N} - \mathcal{D}(S)$. Assume now that node S meets node U. According to protocol **2H**, message M is delivered to node $U \in \mathcal{D}(S)$. If node U later on meets node V, the forwarding condition is satisfied: in fact, $|65 - 59| = 6 < 10 = \lambda$, which implies that $vcw(U, V, \lambda) = 1$. Since $V \in \mathcal{N} - \mathcal{D}(S)$, this proves that protocol **2H** can deliver message M also to unintended nodes.

Finally, protocol **R2** aims at achieving an optimal tradeoff between precision and coverage by tuning parameter λ'. In particular, the following proposition states that setting $\lambda' = \lambda/2$ guarantees maximum precision for protocol **R2**. In the following, we call the version of **R2** with $\lambda' = \lambda/2$ protocol **E2** (where **E** stands for *exact*), to emphasize the fact that under this protocols the message is delivered only to nodes in $\mathcal{D}(S)$.

Proposition 1: If $\lambda' = \lambda/2$, then protocol **R2** achieves maximum precision, i.e., $Prec(\mathbf{E2}) = 1$.

Proof: To prove the claim, it is sufficient to show that messages M can be delivered only to nodes in set $\mathcal{D}(S)$. To show this, we first observe that M can be delivered to a node V either: 1) directly from node S; or 2) from another node U which received M directly from S (two-hops forwarding). We prove that the condition $\mathbb{C}$ ="node $V \in \mathcal{D}(S)$" holds in both cases. In the first case, the specification of the forwarding protocol requires that $V \in \mathcal{D}(S)$. In the second case, the specification of protocol **E2** requires that node U can act as forwarder if and only if it satisfies condition $vcw(U, S, \lambda/2) = 1$. This implies that, for any interest dimension i, we have $|v_i - s_i| \leq \lambda/2$. In order for M to be delivered to node V, protocol **E2** requires that $vcw(V, U, \lambda/2) = 1$, which, in turn, implies $|u_i - v_i| \leq \lambda/2$ for any i. Since $|u_i - s_i| \leq \lambda/2 \wedge |v_i - u_i| \leq \lambda/2$ imply that $|v_i - s_i| \leq \lambda$, and the inequality holds for any i, we have that $vcw(V, S, \lambda) = 1$, i.e., $V \in \mathcal{D}(S)$ and the proposition follows. □

Notice that protocol **R2** achieves maximum precision for any value of λ' strictly smaller than $\lambda/2$. However, with such settings of λ' we would have restricted forwarding opportunities, while not increasing precision with respect to setting $\lambda' = \lambda/2$. For this reason, values of λ' strictly smaller than $\lambda/2$ (where λ is assumed here to be an even value for simplicity) are not useful in practice.

5 Forwarding and Privacy

For definiteness, in the following we assume that a user's interest is defined in a fixed range —from 1 to 100—, reflecting the fact that a participant can be maximally (100) or minimally (1) interested in receiving messages about a specific topic. Table 2 reports a possible *Alice's interest profile*, with a *degree of interest* expressed for each topic. Similarly, Bob's interest profile is also shown in Table 2. We recall that the general idea of interest-based forwarding [16] is that when Bob and Alice meet, they should be able to share messages in their respective buffers if they find their interest profiles similar enough, guided by the principle that a piece of information which is relevant for Alice might be interesting for any other individual (e.g., Bob) with similar interests.

Notice that, similar to other social-aware forwarding approaches introduced in the literature, interest-casting introduces several problems concerning user privacy, if adequate counter-measures are not undertaken. In fact, the interest-casting approach presented in [16] assumes that Alice and Bob exchange their *plain* interest profiles, thus revealing to the other party very sensitive personal information.

TABLE 2
Alice's and Bob's interest profiles

User	Cinema	Book	Music	...	Car
Alice	45	32	69	...	10
Bob	30	65	71	...	88

Examples of attacks a malicious Bob may perform are the following:

- He may discover the degree of Alice's interest in each topic.
- He may download all messages in Alice's buffer by first acquiring Alice's profile in a first interaction, then creating a false identity (*Sybil attack*) with an artificially created interest profile resembling Alice's one, and then interacting again with Alice through the faked identity.
- He may reveal information obtained from Alice's interest profile to another user *(Collusion Attack)*.

Hence, it is a common opinion that user interest profiles should be kept private, and only minimal information must be disclosed by a user when profiles are matched for similarity computation. However, it

is important to observe that there exists an inherent tradeoff between the need of preserving privacy, and the efficiency/accuracy of information forwarding in the network. This tradeoff, which will be carefully investigated in the remainder of this paper, is due to the fact that interest-casting dictates that messages are circulated *only between network members sharing similar interests*. Thus, trivial privacy-preserving solutions in which no information about interest profiles is exchanged, and message forwarding proceeds in an epidemic fashion, are not acceptable if effective interest-casting is the design goal. Summarizing, a certain leakage of information regarding interest-profiles is inevitable, given the need of designing an effective interest-cast protocol.

Remark: In this paper, we consider the problem of preserving privacy of the user interest profiles. The problem of ensuring security of the information exchanged in the network by means of interest-casting is largely orthogonal to the problem of ensuring privacy of interest-profiles considered herein. Nevertheless, a couple of observations are in order. In many application scenarios, it is reasonable to assume that the interest-casted information is not sensitive, such as information about events in a community, etc. In case the interest-casted information is instead sensible, data encryption mechanisms should be used to protect it. It is interesting to observe that, even in this scenario, some of the forwarding protocols considered in this paper (more specifically, **DD** and **E2**) satisfy the property that information is exchanged only between users whose profile is guaranteed to be similar to the one of the user who generated the content. Thus, if we assume a trust model in which a mutual trust relationship is established whenever two users share common interests, standard techniques can be used to setup a secure channel between trusted parties.

6 Privacy-preserving interest-casting

To address the fundamental tradeoff between privacy and forwarding accuracy, we present a privacy-preserving version of interest-casting based on a protocol used to solve the well-known "Millionaire's Problem" introduced by Yao [18], which is an instance of secure two-party computation.

We recall that the goal of the "The Millionaire's Problem" is to compare two numbers, i and j, and to discover whether:

$$i \leq j \qquad \text{or} \qquad i > j \tag{1}$$

However, this comparison must not leak out any information about the values of i and j to the other party: if Alice holds i and Bob j, at the end of the protocol's execution Alice knows only whether Bob holds a number larger than i, and not the actual value of j (similarly for Bob). Notice that *there is* a privacy leakage after the protocol execution. Namely, at the end of the protocol's execution both Alice and Bob know whether the other party holds a larger or smaller number than the own value. However, this privacy leakage is unavoidable, if the goal is jointly computing condition (1) above.

Since its introduction in [18], the "Millionaire's Problem" has been widely studied in the literature, mainly with the goal of reducing the computational complexity of the cryptographic primitives used in the protocol [24]. Reducing computational complexity is especially important in the scenario at hand, where the protocol should be executed on mobile devices. Recently, efficient solutions to the "Millionaire's Problem" have been proposed. For instance, in [25], [26] the authors propose protocols for solving the "Millionaire's Problem" based on asymmetric cryptography, e.g., *RSA*. In addition, in [25] the authors also propose a version of the protocol that uses symmetric keys and real numbers. Computational time evaluations, obtained using an old Pentium III/450Mhz, prove that the hardware of recent mobile devices (with CPU speed up to 1Ghz and above) is able to efficiently run these solutions to the "Millionaire's Problem".

6.1 The Millionaire's Problem and Interest-cast

In this section, we show how a protocol to solve the "Millionaire's Problem" can be modified to securely compute the condition below:

$$|i - j| \leq \lambda \ . \tag{2}$$

In inequality (2), i represents Alice's interest in a specific topic, and j Bob's interest in the same topic. By repeating the protocol on each topic of the interest profile, Alice and Bob can securely compute the similarity metric $vcw(Alice, Bob, \lambda)$ defined in Section 3.

Notice that, similarly to the original "Millionaire's Problem", there is an unavoidable privacy leakage caused by the execution of the protocol. Namely, at the end of one iteration of the protocol execution, Alice knows whether j is inside or outside the interval $[i - \lambda, i + \lambda]$ (similarly for Bob). Thus, if the protocol to jointly compute inequality (2) is repeated on each topic of the interest profile, at the end of the execution the information of whether j belongs to interval $[i - \lambda, i + \lambda]$ is known for each possible topic of the interest profile. To reduce this privacy leakage, and to limit the impact of Sybil attacks – see next section, we propose that, when Alice and Bob meet, they *estimate* the similarity of their profiles based on a *random subset* of topics of *fixed cardinality* k, with $k < m$. More specifically, upon encounter Alice and Bob compute an *estimated* similarity metric $vcw_e(Alice, Bob, \lambda)$, where $vcw_e(Alice, Bob, \lambda) = 1$ if and only if $|x_i - y_i| \leq \lambda$ for each $i \in \mathcal{I}$, where $\mathcal{I}$ is the set of indexes of the topics in the random set, with $|\mathcal{I}| = k$. The forwarding/message delivery decision in any of the three considered forwarding protocols is then taken based on the outcome of the vcw_e similarity metric, instead of the original vcw metric.

It is important to observe that using metric vcw_e to govern forwarding decisions introduces inaccuracies in the message delivery process. In particular, it is possible, even in case of protocols **DD** and **E2**, to deliver a copy of

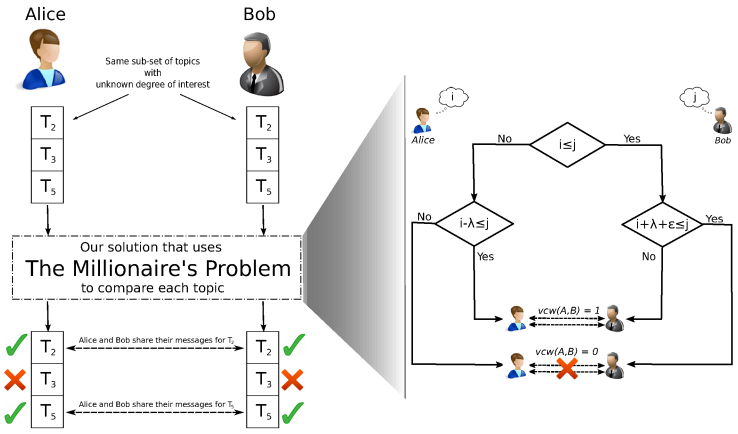

Fig. 1. Privacy-Preserving Interest-based Forwarding

the message to unintended recipients. In fact, it is easy to see that, while $(vcw(A, B, \lambda) = 1) \Rightarrow (vcw_e(A, B, \lambda) = 1)$, the opposite implication does not hold. In other words, due to the reduced number of topics on which the similarity metric is computed, it is possible to erroneously estimate as similar two individuals whose complete profiles do not satisfy the similarity metric. As the value of k increases, the likelihood of erroneous similarity estimation decreases, but privacy leakage increases. This inherent tradeoff between forwarding accuracy and privacy leakage will be thoroughly investigated in the remainder of this paper.

The protocol we propose to securely verify condition (2) on a single topic is reported in Figure 1 – right. The protocol consists in first verifying condition $i \leq j$ using an efficient protocol for solving the "Millionaire's Problem", such as those presented in [25], [26]. Depending on the outcome of the first condition (which is known to both parties), another condition is verified: $i - \lambda \leq j$ if the outcome of the first condition was negative, or $i + \lambda + \epsilon \leq j$ otherwise, where ϵ is an arbitrarily small positive number known to both parties. Notice that also the value of λ is assumed to be known to both parties. Depending on the outcome of this second condition, a common decision on the value of the vcw metric is taken. It is immediate to see that, after the protocol execution, $vcw(A, B, \lambda) = 1$ if and only if $|i-j| \leq \lambda$ (up to the small approximation introduced by the constant ϵ).

The complete protocol for estimating metric vcw_e using a random subset of k topics, which we name PPIF (Privacy-Preserving Interest-based Forwarding), is reported in Figure 1–left. Alice initiates the protocol communicating Bob her intention to jointly compute similarity. Bob chooses uniformly at random a set of k topics in his interest profile, and communicates the chosen subset to Alice. Then, Alice and Bob verify similarity on each of the chosen topics using the protocol of Figure 1–right. The value of the jointly computed similarity metric vcw_e is then obtained by performing a logical AND operation between the similarity values obtained in each topic. Notice that, in order to reduce privacy leakage, we assume that some form of authentication is used to prevent Alice from repeatedly executing the protocol with Bob in a limited time frame. For instance, we can assume that only a single interaction between Alice and Bob is permitted within an hour, a day, etc.

We have recently shown [27] that the PPIF protocol can be implemented in mobile platforms with very reasonable running times (less than $5secs$ when vcw_e is computed using 4 topics).

As it will be carefully discussed in Section 7, the choice of letting Bob, instead of Alice, randomly select the subset of topics used to compute vcw_e is motivated by the need of reducing the impact of Sybil and Collusion attacks.

7 Analysis

In this section, we discuss the security properties of our privacy-preserving interest-casting protocol and analyze its robustness against different types of attacks.

7.1 Security properties

The basic security properties of our proposed protocol are directly derived from the properties of the underlying protocol used to solve the "Millionaire's Problem". Before proceeding further, we need to define the possible attacker models.

Within the context of secure two-party computation, two attacker models are typically considered:

- *semi-honest model*: in this model, the attacker is assumed to behave according to the protocol specifications, with the exception that she/he keeps track of all the intermediate computations with the purpose of trying to derive the other party's input. This model is also named *honest but curious* model in the literature;
- *malicious model*: in this model, the attacker may behave arbitrarily, including refusing to participate in the protocol, substituting an input with an arbitrary value, prematurely aborting the protocol, etc.

The protocols for solving the "Millionaire's Problem" introduced in the literature, including [25], [26], are shown to be *secure* against both semi-honest and malicious attackers. The word "secure", in the context of secure two-party computation, means that at the end of

the protocol execution, both parties only know the outcome of the function evaluation (verification of condition $i \leq j$ in our case), with the minimal privacy leakage that comes out from the outcome of the function used. Furthermore, as typical in secure two-party computation, there is no other way to avoid that Bob does not send to Alice the outcome of the function at the end of the protocol. Thus, if Bob is a malicious attacker[4], he could end up the protocol before communicating to Alice the result of the function computation. In such a case, though, Bob would still know only the minimal information about Alice's input that can be derived from the outcome of the function.

Notice that, differently from the "Millionaire's Problem", in the PPIF protocol, several computations of different functions (conditions on topic values) – namely, $2k$ – must be performed to jointly compute the metric vcw_e. These repeated executions of the underlying "Millionaire's" protocol might impair the security properties of the PPIF protocol. However, we have verified [27] that the $2k$ logical interactions in the PPIF protocol can be encoded into a single actual interaction using the FairPlay framework for secure two-party computations [28]. This framework is also robust w.r.t. the previous attack models. We can then conclude that the PPIF protocol we developed is then secure against both semi-honest and malicious attackers.

7.2 Privacy preservation

As commented in Section 5, there is an inherent privacy leakage vs. forwarding accuracy tradeoff in privacy-preserving interest-casting. To quantify this tradeoff, we introduce a privacy preservation metric based on the information-theoretic notion of entropy introduced by Shannon [29] (see also [30], [31] for other approaches using entropy in security). The Shannon entropy is a measure of the average information content that is missing when the value of a random variable is not known.

In our setting, Alice's interest profile, from Bob's perspective, can be considered as a random variable. Thus, the notion of entropy can be used to quantify Bob's uncertainty about the value of Alice's profile.

More specifically, the random variable of interest is an m-dimensional random variable $X = (X_1, \ldots, X_m)$, where all random variables X_i have the same support $[1, max]$. In order to simplify the presentation, we assume in the following that random variables X_i are *mutually independent*. Although we acknowledge that in practice the interest values in different topics can be correlated, we retain the independence assumption here to simplify the definition of the introduced privacy leakage metric.

By definition [29], the *bit entropy* of a random variable Y with possible values $\{y_1, \ldots, y_n\}$ is defined as:

$$H[Y] = -\sum_{i=1}^{n} p(y_i) \log_2 p(y_i) \ ,$$

where $p(y)$ is the probability mass function of random variable Y.

If Y and Z are independent random variables, we can write [29]:

$$H[(Y, Z)] = H[Y] + H[Z] \ ,$$

from which we obtain:

$$H[X] = \sum_{i=1}^{m} H[X_i] \ .$$

We are now ready to introduce our privacy preservation metric, whose purpose is to quantify the privacy leakage (decrease of entropy) induced by one or multiple executions of the PPIF protocol. Let $X_{initial}$ and X_{after} be the random variables modeling Bob's uncertainty about Alice's interest profile *initially* and *after* a single execution of the PPIF protocol. Notice that, since Bob acquires some knowledge about Alice's profile during PPIF execution, we have in general that $H[X_{after}] \leq H[X_{initial}]$. We can then define the following *privacy preservation* metric:

$$pp(PPIF) = \frac{H[X_{after}]}{H[X_{initial}]} \ .$$

The privacy preservation metric takes values in $[0, 1]$, with 0 indicating that after PPIF execution Bob knows exactly Alice's interest profile (zero privacy preservation), and 1 indicating that after executing PPIF Bob has the same knowledge about Alice's profile he had before executing the protocol (maximal privacy preservation). Defining a privacy metric in the $[0, 1]$ interval is especially important since it allows combining this metric with the forwarding performance metrics of precision and coverage (also defined in the $[0, 1]$ interval), and computing the overall pF score used to rank forwarding protocols in the simulation experiments reported in Section 8.

To make the discussion more concrete, in the following we assume that random variables X_is have uniform distribution in the $[1, max]$ interval. Under this assumption, we have that

$$\begin{aligned} H[X_{initial}] &= \sum_{i=1}^{m} H[X_i] = -\sum_{i=1}^{m} \sum_{j=1}^{max} \frac{1}{max} \log_2 \frac{1}{max} = \\ &= \sum_{i=1}^{m} \log_2 max = m \log_2 max \ . \end{aligned}$$

To quantify $H[X_{after}]$, we start assuming that $k = 1$, and observe that PPIF might have two possible outcomes:

- $vcw_e(Alice, Bob, \lambda) = 0$. In this case, Bob knows that $|i - j| > \lambda$ on a specific topic. Thus, the set of possible values of i is reduced from max to

4. Notice that the role of Alice and Bob in this section is reversed with respect to what reported in Figure 1: Bob (the attacker) initiates the protocol, and Alice chooses the random set of topics to compute the vcw_e metric.

$max - 2\lambda - 1$ in the tested topic. In the other topics, the uncertainty about Alice's value is unchanged. We can then conclude that

$$H[X^0_{after}] = (m-1)\log_2 max + \log_2(max - 2\lambda - 1) \ .$$

– $vcw_e(Alice, Bob, \lambda) = 1$. In this case, Bob knows that $|i-j| \leq \lambda$ on the tested topic, and the set of possible values of i on that topic is reduced from max to $2\lambda + 1$. Similarly to above, the uncertainty in the other topics is unchanged, and we can write:

$$H[X^1_{after}] = (m-1)\log_2 max + \log_2(2\lambda + 1) \ .$$

We also notice that event $vcw_e(Alice, Bob, \lambda) = 0$ occurs with probability equal to $\frac{max-2\lambda-1}{max}$ (under the assumption that $\lambda \ll max$), while event $vcw_e(Alice, Bob, \lambda) = 1$ occurs with probability $\frac{2\lambda+1}{max}$. Thus, the expected value of the privacy preservation metric with $k = 1$ (normalized in order to have it between 0 and 1) amounts to:

$$E_m[pp(PPIF_1)] = \frac{\frac{max-2\lambda-1}{max} \cdot H[X^0_{after}] + \frac{2\lambda+1}{max} \cdot H[X^1_{after}]}{m \log_2 max}$$

where $PPIF_1$ indicates that the protocol is executed with $k = 1$.

To compute the expected privacy preservation resulting from a single execution of the PPIF protocol with $k > 1$, we start observing that the entropy of random variable X_{after} when $vcw_e(Alice, Bob, \lambda) = 0$ is:

$$H[X^0_{after}] = \log_2(max^m - (2\lambda + 1)^k) \ ,$$

which follows from the observation that, under the assumption that the m interest values are independent and uniformly distributed in $[1, max]$, all the max^m possible values of Alice's profile are equiprobable and, out of those, only the ones in which inequality $|i - j| \leq \lambda$ is verified for all k considered topics are not possible, given the outcome $vcw_e(Alice, Bob, \lambda) = 0$. In fact, in the PPIF protocol it is Alice who knows the outcome of the test on each specific topic and performs the logical AND operation to compute vcw_e, and then send to Bob the outcome of the logical AND operation.

On the other hand, it is easy to see that the entropy of random variable X_{after} when $vcw_e(Alice, Bob, \lambda) = 1$ is

$$H[X^1_{after}] = (m-k)\log_2 max + k \log_2(2\lambda + 1) \ .$$

We then observe that the probability of event $vcw_e(Alice, Bob, \lambda) = 0$ when vcw_e is computed using k topics is $1 - \left(\frac{2\lambda+1}{max}\right)^k$, and that the probability of the complementary event $vcw_e(Alice, Bob, \lambda) = 1$ equals $\left(\frac{2\lambda+1}{max}\right)^k$. Thus, we can conclude with the following theorem:

Theorem 1: The expected privacy-preservation when protocol PPIF is executed using $k > 1$ topics to compute similarity is:

$$E_m[pp(PPIF_k)] =$$

$$= \frac{\left(1 - \left(\frac{2\lambda+1}{max}\right)^k\right) \cdot H[X^0_{after}] + \left(\frac{2\lambda+1}{max}\right)^k \cdot H[X^1_{after}]}{m \log_2 max} \ ,$$

where

$$H[X^0_{after}] = \log_2(max^m - (2\lambda + 1)^k) \ ,$$

and

$$H[X^1_{after}] = (m-k)\log_2 max + k \log_2(2\lambda + 1) \ .$$

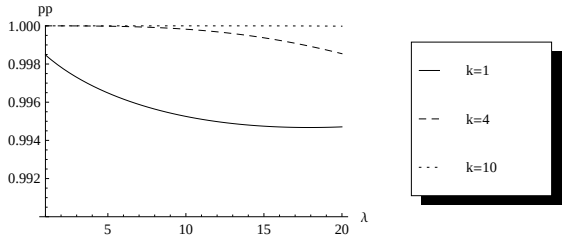

Fig. 2. Value of the expected privacy preservation metric for increasing values of λ and different values of k. Parameter max is set to 100 and m is set to 15.

As shown in Figure 2, the expected privacy preservation of protocol PPIF is very close to 1 for reasonable setting of the parameters, due to the fact that event $vcw_e(Alice, Bob, \lambda) = 0$, which leads to negligible privacy leakage, occurs with high probability. For instance, when $max = 100, \lambda = 10$ and $k = 4$, we have that event $vcw_e(Alice, Bob, \lambda) = 0$ occurs with probability 0.998. It is also interesting to note that the value of k strongly influences the probability of event $vcw_e(Alice, Bob, \lambda) = 1$, i.e., of a successful message forwarding. For instance, $Prob(vcw_e(Alice, Bob, \lambda) = 1) = 0.002$ when $k = 4$, indicating that only 2 out of 1000 forwarding opportunities are exploited in average. However, by setting $k = 1$ we have $Prob(vcw_e(Alice, Bob, \lambda) = 1) = 0.21$, which increases the average number of exploited forwarding opportunities to 1 out 5. The strong influence of parameter k on the message forwarding process is confirmed by the simulation results reported in Section 8.

It is also useful to define the notion of *worst-case* privacy preservation to investigate the privacy leakage vs. forwarding accuracy tradeoff. Worst-case privacy preservation is computed assuming the event leading to the highest privacy leakage (namely, event $vcw_e(Alice, Bob, \lambda) = 1$) occurs. It is easy to see that the worst-case privacy preservation of protocol PPIF using $k \geq 1$ topics to compute metric vcw_e amounts to

$$WS[pp(PPIF_k)] = \frac{(m-k)\log_2 max + k \log_2(2\lambda + 1)}{m \log_2 max} \ .$$

The value of $WS[pp(PPIF_k)]$ as λ increases from 1 to 20 for different values of k is reported in Figure 3. As seen from the plot, worst-case privacy can be augmented by increasing λ and/or decreasing k. In particular, while decreasing k has a clear effect on the pp metric, increasing the value of λ only marginally improve privacy preservation beyond a certain threshold (e.g., around

5 for $k = 4$). However, as we will discuss below and in Section 8, relatively high values of λ and relatively small values of k negatively affect the accuracy of the forwarding process. Clearly, privacy is also increased for increasing values of m.

Overall, the results presented in this section can be used by the protocol designer to identify the best tradeoff between privacy preservation and forwarding accuracy as a function of the protocol parameters. An example of this usage is the computation of the pF score in the simulation experiments to identify the best performing protocol – see Section 8.

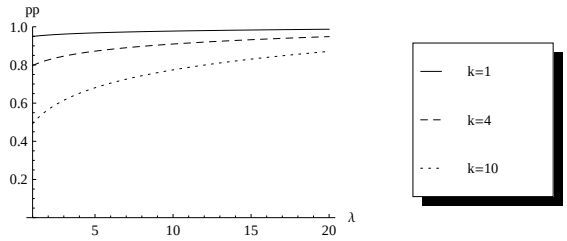

Fig. 3. Value of the worst-case privacy preservation metric for increasing values of λ and different values of k. Parameter max is set to 100 and m is set to 15.

7.3 Forwarding accuracy

As we have already observed, using the vcw_e metric instead of the complete vcw one to compute similarity between interest profiles introduces inaccuracies in the forwarding process. In particular, the following event, which we call *mis-forwarding* and denote MS, can occur with non-zero probability: "$(vcw_e(Alice, Bob, \lambda) = 1) \wedge (vcw(Alice, Bob, \lambda) = 0)$". Mis-forwarding should be avoided to preserve confidentiality of information (if shared data is intended to circulate only amongst users with similar profiles), and to avoid spamming of undesired information to un-interested users more in general. In what follows we estimate the probability of event MS as a function of k.

Let $Ct(A, B)$ denote the number of topics in Alice's and Bob's profile for which condition $|i - j| \leq \lambda$ is satisfied. We can write:

$$P(MS) = \sum_{t=k}^{m-1} P(MS|Ct(A,B) = t)P(Ct(A,B) = t) \ .$$

Notice that the summation above starts from $t = k$, since if Alice and Bob have less than k topics in common, event MS cannot occur. Similarly, if $t = m$ then $vcw(Alice, Bob, \lambda) = 1$, and MS cannot occur as well.

If we assume that interests in each topic are uniformly distributed in the $[1, max]$ interval, the similarity event on a single topic can be considered as a Bernoulli trial, and we can compute $P(Ct(A, B) = t)$ as follows:

$$P(Ct(A,B) = t) \approx \binom{m}{t} \left(\frac{2\lambda+1}{max}\right)^t \cdot \left(1 - \frac{2\lambda+1}{max}\right)^{(m-t)} \ ,$$

where the approximation is due to the fact that the success probability in each of the m Bernoulli trials is less than $(2\lambda + 1)$ if the value of i or j is close to the border of the $[1, max]$ interval. It is easy to see that the above approximation is very accurate whenever $\lambda \ll m$.

Conditioned on event $Ct(A, B) = t$, the probability of event MS can be computed as follows:

$$P(MS|Ct(A,B) = t) = \frac{\binom{t}{k}}{\binom{m}{k}} \ ,$$

where, we recall, $k \leq t \leq m - 1$.

The probability of mis-forwarding for increasing values of k and different values of λ is reported in Figure 4. As seen from the figure, setting $k = 4$ already ensures a negligible probability of mis-forwarding, while at the same time providing good privacy-preservation properties (recall Figure 3).

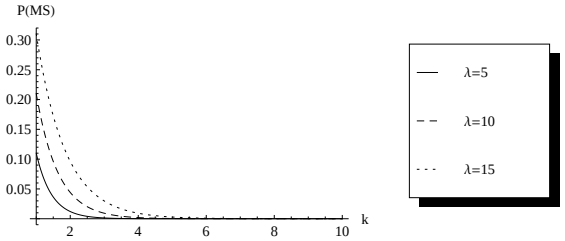

Fig. 4. Probability of mis-forwarding for increasing values of k and different values of λ. Parameter max is set to 100 and m is set to 15.

7.4 Sybil attack

In order to acquire more information about Alice's interest profile, an attacker (Bob) could forge multiple identities and perform repeated interactions with Alice. In this section, we estimate the number of identities Bob has to forge in order to perform a successful Sybil attack.

To this end, we start defining two notions of Sybil attack:

- *weak* Sybil attack: in this case, the attacker's goal is to perform a successful interaction with Alice. In other words, the attacker's goal is to compute a profile such that the metric vcw_e computed against Alice's profile returns 1 (i.e., at the end of a successful attack, the attacker knows the Alice's profile up to λ).
- *strong* Sybil attack: in this case, the attacker's goal is to discover Alice's exact profile. Clearly, this implies complete privacy leakage. This attack is much more valuable than the other for a potential attacker, since a user's profile can be considered as an asset many companies, organizations like political parties, etc., might be interested in.

The number of identities required to successfully perform a weak Sybil attack when $k = 1$ can be estimated noticing that the problem under investigation is similar to the well-known coupon collector's problem [32]: since Alice chooses the topics on which the vcw_e metric is computed uniformly at random, each time a Bob's fake identity interacts with Alice, this is equivalent to

randomly extracting a coupon (topic) among a set of m possible coupons. In order to perform a successful attack, Bob needs to extract all the m coupons. The expected number of tries in order to collect m coupons is given by [32]: $m \cdot H_m$ where $H_m \sim \log m$ is the m-th Harmonic number. In order to perform a weak Sybil attack, Bob has to interact several times with Alice on each topic, each time using a different value as the own (fake) interest in the topic, with the purpose of disclosing Alice's value on that topic. The number of interactions needed to discover the value i of Alice on each topic can be computed as follows. With $\frac{max}{2\lambda+1}$ interactions, Bob can discover in which of the mutually disjoint intervals of length $2\lambda+1$ value i lies in. Thus, $s_W = \frac{max}{2\lambda+1}$ interactions are needed on each of the m topics to fully disclose Alice's profile, where subscript W stands for "weak". To sum up, the total number of identities needed to successfully perform a weak Sybil attack can be estimated by observing that the problem at hand is equivalent to the s_W-coupon collector's problem, where the collector's goal is collecting s copies of each coupon. From [33], we get

$$E_{wS}[Id] = m \cdot \log m + (s_W - 1)m \log\log m + O(m) \ . \quad (3)$$

For the strong attack, one can consider $\log(2\lambda+1)$ additional interactions (with binary search) to exactly identify the value of i within the selected interval. Thus, $s_S = \frac{max}{2\lambda+1} + \log(2\lambda+1)$ interactions are needed on each of the m topics to fully disclose Alice's profile, where subscript S stands for "strong". The expected number of identities needed to successfully perform a strong Sybil attack can then be computed according to equation (3), with s_W replaced with s_S.

The expected number of identities needed to successfully perform a weak or strong Sybil attack to the PPIF protocol when $k = 1$ for increasing values of m is reported in Figure 5. For instance, when $m = 15$, the attacker needs approximately 54 identities to perform a weak Sybil attack, while he/she needs about 59 identities to perform a strong Sybil attack. Unfortunately, extending the above analysis to the case $k > 1$ is highly non-trivial, since in this case the number of new coupons that are collected at each interaction (which is either 0 or 1 when $k = 1$) can take any value between 0 and k, leading to an explosion of the number of possible ways of collecting all the m coupons, which hinders analytical derivation.

It must be noticed that in the above analysis we assumed that it is always the attacker who starts the protocol with Alice, giving to Alice the possibility of randomly choosing the topics. In practice, it might happen that Alice starts the interaction with one of Bob's sybils, which would give the attacker the opportunity of choosing the topics. Thus, the one reported above must be considered as the worst case for the attacker.

The analysis above applies also to the situation in which the forgery of identities is not possible, e.g. in presence of strong authentication mechanisms. In this situation, an alternative attack is the collusion attack, in which multiple malicious users cooperate and share knowledge they acquire on Alice's profile. Multiple malicious users are equivalent to multiple identities in a Sybil attack, and the analysis above can be readily applied also to deal with collusions attacks.

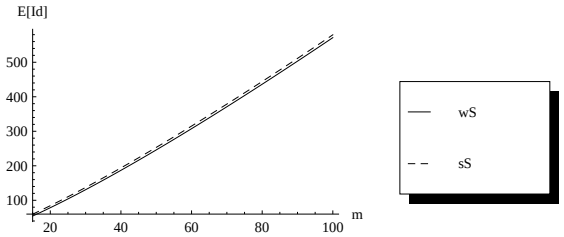

Fig. 5. Expected number of identities needed to successfully perform a weak or strong Sybil attack for increasing values of m. Parameters are $max = 100$ and $\lambda = 10$.

8 Simulations

In this section, we report the results of the simulations we have performed to better investigate the privacy preservation vs. forwarding accuracy tradeoff in interest-casting. The simulations are based on the MIT reality mining mobility trace [34], which collects mobility traces of 97 users holding a Bluetooth device from July 2004 to April 2005. The trace was generated by periodically performing a BT scan operation, and by recording the ID of neighboring devices and time of contact. Unfortunately, the trace does not contain any information about user interest profiles. To fill this gap, we decided to run a survey at our Institute (the "Istituto di Informatica e Telematica" of the Italian National Research Council), asking 97 volunteers to fill an online, anonymous form. The form contained 15 topics (e.g., "Books", "Music", "Sport", etc.); for each topic, we asked the user to indicate an interest value in the range [1,100]. Then, we associated each generated interest profile to a randomly chosen user in the MIT trace.

In order to perform a number of significant experiments, we pre-processed user interest profiles, with the purpose of selecting a representative subset of the interests where the user population display a sufficient degree of similarity. In fact, considering the limited number of users (97) in the MIT trace, and the relatively large number of topics (15) in the user profile, if no pre-processing is performed it is likely that the set $\mathcal{D}(S)$ of relevant destinations for a certain source node S is empty (no users with profile similar to S on all the topics). Clearly, if $\mathcal{D}(S) = \emptyset$ no other node in the network should receive the message generated by S, and the notion of interest-cast is not meaningful.

To avoid the above described situation, we have identified six topics in which user interests are highly concentrated on relatively few values (typically, either very high or very low). These topics are: "Boats", "Books", "Cars", "Environment", "Music", and "Traveling". Since user interests in these topics are highly concentrated on

relatively few values, it is more likely that set $\mathcal{D}(S)$ contains at least one node as desired. The simulation results reported in the following are then obtained setting $m = 6$ and varying the number k of topics used to compute the estimated metric vcw_e, where the six topics in the user interest profiles are the ones mentioned above.

A simulation experiment consists in randomly selecting a source node S, and in initially computing the set $\mathcal{D}(S)$ of relevant destinations based on user interest profiles. The value of λ used to compute the similarity metric is $\lambda = 20$. If set $\mathcal{D}(S)$ is empty, the simulation is discarded for the reason explained above. Otherwise, an interest-cast message is generated at node S with TTL equal to the duration of the data trace, and is propagated in the network according to one of the three forwarding protocols defined in Section 4, namely, **DD**, **2H**, and **E2**. Then, the following metrics are computed:

- *coverage*, as defined in Section 3;
- *precision*, as defined in Section 3;
- *delay*, defined as the average delay with which the message is received (computed only for messages received by nodes in set $\mathcal{D}(S)$).

The results presented in the following refer to 97 simulation experiments, corresponding to generating a message at each node in the network. In 42 out of 97 experiments, the condition $|\mathcal{D}(S)| > 0$ was satisfied, and performance of the different forwarding protocols evaluated.

For the purpose of comparison, we implemented also the SANE interest-cast protocol introduced in [16], which is not privacy aware (interest profiles are plainly exchanged between users). There are two differences between SANE and the protocols introduced here. First, the metric used for computing interest similarity is the cosine metric, instead of vcw. Second, since privacy is not a concern in SANE, when a message is propagated multihop in the network, the profile of the possible forwarder/destination node is compared with the profile of the *source* node which generated the message. In other words, a message circulating in the network carries also the interest profile of the source node, which is not the case in our multihop protocols due to the need of preserving privacy. We have implemented two forwarding rules for SANE, corresponding to **DD** and **2H** forwarding.

In order to make the comparison between SANE and our multi-hop protocols the fairest possible, we have proceeded as follows. First, we have computed the average size of the set $\mathcal{D}(S)$ of intended destinations on the 42 selected source nodes, where set $\mathcal{D}(S)$ is computed using the vcw similarity metric. This average size turned out to be 6.4. For the same set of 42 source nodes, we have then computed the size of the set $\mathcal{D}_c(S)$ of intended destinations using the cosine metric, using different similarity thresholds γ. Finally, we have selected the value of γ corresponding to a size of the set $\mathcal{D}_c(S)$ as close as possible to 6.4. This value turned out to be $\hat{\gamma} = 0.99$, for which the average size of the sets $\mathcal{D}_c(S)$ is 5.62.

The coverage, precision, and delay of the different forwarding protocols obtained when k varies from 1 to 6 are reported in Figure 6. The figure also reports 95% confidence intervals. First, we observe that coverage with **DD** protocol is independent of k. This comes from the fact that in **DD** it is only the source node that can deliver the message to a node in $\mathcal{D}(S)$, and that when S meets a node U in $\mathcal{D}(S)$ it always delivers the message to U independently of the value of k. In fact, $(vcw(U,S) = 1) \Rightarrow (vcw_e(U,S) = 1)$, independently of the value of k and of the specific topics used to compute the vcw_e metric. On the other hand, the precision of **DD** increases with k, since a higher value of k results in a lower occurrence of false positives in the computation of the vcw_e metric (recall also Figure 4). Maximum precision of 1 is achieved when $k = 6$, since in this case $vcw \equiv vcw_e$, and in protocol **DD** only the source node is allowed to deliver the message to nodes in $D(S)$ – recall Section 4.1.

Protocols **2H** and **E2** stand at the opposite ends of the coverage vs. precision tradeoff. Protocol **2H** has the best coverage performance ($\approx 87\%$ when $k = 1$), thanks to the two-hops propagation of the interest-cast message. Coverage has a decreasing trend with k, due to the fact that more copies of the message circulate in the network with lower values of k. However, the better coverage with low values of k is paid in terms of decreased precision, which is very low for values of $k \leq 4$.

Conversely, protocol **E2** provides the best precision performance, with optimal precision of 1 obtained with $k = 6$. The good precision performance of the **E2** protocol is due to the fact that a lower value of $\lambda' = \lambda/2 = 10$ is used to compute the estimated similarity metric vcw_e. However, the good precision performance is paid in terms of coverage, which is always lower than that provided by protocol **2H**: the decreasing trend of coverage with increasing k with the **E2** protocol is more pronounced than with the **2H** protocol. When $k > 3$, **E2** coverage is worse than that provided by protocol **DD**, which does not exploit multi-hop message propagation.

Since SANE uses the cosine similarity metric for message forwarding/delivery, its performance is not influenced by the choice of parameter k. In terms of precision, SANE is clearly optimal, since the plain interest profile of the message source is propagated in the network jointly with the message (compromising privacy). In terms of coverage, SANE provides intermediate performance, covering 42% of the intended destinations with **DD** forwarding and 56% of the intended destinations with **2H** forwarding.

To have a more comprehensive understanding of the coverage vs. precision tradeoff with the various protocols, we have also computed the *F-score*, which is formally defined as follows [23]:

$$F(\mathbf{F}) = 2 \cdot \frac{Prec(\mathbf{F}) \cdot Cov(\mathbf{F})}{Prec(\mathbf{F}) + Cov(\mathbf{F})} \ ,$$

where $\mathbf{F}$ is the forwarding protocol at hand. The F-score

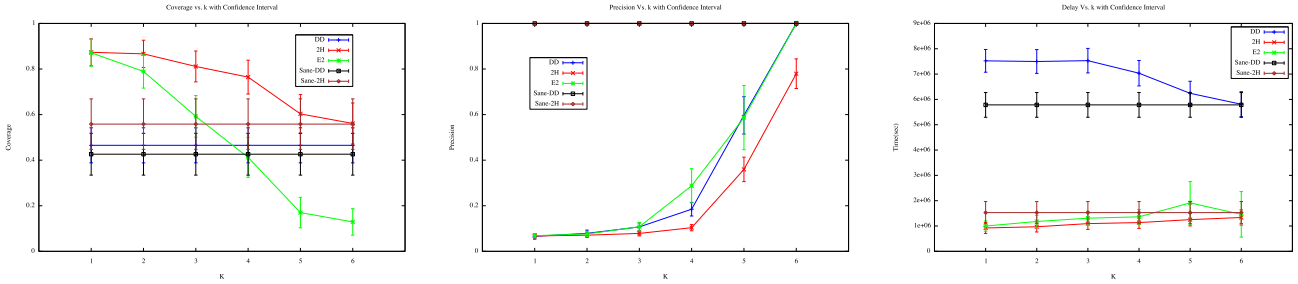

Fig. 6. Coverage (left), precision (center), and delay (right) of the different forwarding protocols when $k = 1, \ldots, 6$.

of the various protocols for increasing values of k is reported in Figure 7. As seen from the plot, the best F-score is provided by SANE with **2H** forwarding, which, however, is not privacy preserving. Amongst privacy preserving protocols, the best F-score is provided by protocol **2H** when $k = 6$. More in general, the best F-score is provided by protocol **DD** when $k = 1, 5$, by protocol **E2** when $k = 2, 3, 4$, and by protocol **2H** for $k = 6$.

Notice, though, that the F-score metric does not account for the privacy level provided by the different forwarding protocols. In order to obtain a single metric accounting also for privacy, we have defined a privacy-preserving version of the F-score metric, which we call *pF-score* and is defined as follows:

$$pF(\mathbf{F}) = 3 \cdot \frac{Prec(\mathbf{F}) \cdot Cov(\mathbf{F}) \cdot WS[pp(\mathbf{F})]}{Prec(\mathbf{F}) + Cov(\mathbf{F}) + WS[pp(\mathbf{F})]} \ ,$$

where $WS[pp(\mathbf{F})]$ is the worst-case privacy-preservation metric of protocol $\mathbf{F}$ which, similarly to coverage and precision, is also defined in the $[0, 1]$ interval. The pF-score of the three protocols for increasing values of k is reported in Figure 8. Notice that, apart from the relatively lower values of the pF-score with respect to the F-score, the relative performance of the various protocols is preserved. In particular, the best pF-score is also achieved when protocol **2H** is used with $k = 6$. Notice finally that the plot of the SANE protocols is not reported, since the privacy preservation metric of SANE is 0 independently of the forwarding strategy. Hence, SANE has pF-score of 0.

It is important to observe that the pF-score as defined above is based on the assumption that, from the application designer's viewpoint, both coverage, precision, and privacy have the same relative importance. In case the three performance metrics have *different* relative importance from the designer's viewpoint, the notion of pF-score defined above can be easily extended to account for different weights of the three performance metrics.

Concerning delay, protocols **2H** and **E2** have similar performance, and they both provide considerably lower delay than that provided by protocol **DD**, which does not exploit multi-hop forwarding of information (see Figure 6). The reduced delay provided by multi-hop forwarding is visible also comparing the delay of SANE with **DD** and **2H** forwarding.

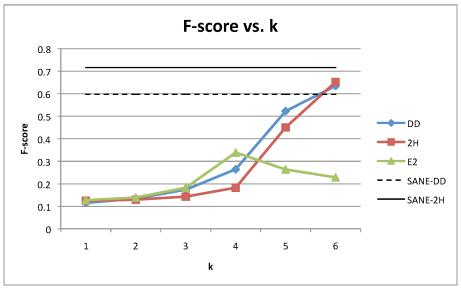

Fig. 7. F-score of the different forwarding protocols for increasing values of k.

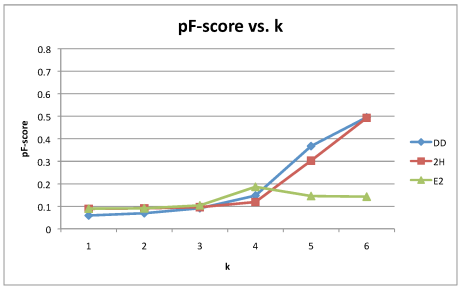

Fig. 8. pF-score of the different forwarding protocols for increasing values of k.

To summarize, simulation results indicate that protocols **2H** and **E2** can be used to address different needs of the interest-casting application: if the designer's goal is to cover as many interested users as possible, protocol **2H** should be preferred. On the other hand, if the designer's goal is to deliver information only to really interested users, so to avoid *spamming* of information within the opportunistic network, then protocol **E2** is the best choice.

9 Conclusion - Future Work

In this paper, we have studied for the first time the fundamental tradeoff between privacy preservation and application-level performance in a specific mobile social networking application, namely, interest-casting. We have introduced three forwarding protocols based on a privacy-preserving detection of users sharing similar interests, two of which exploit multi-hop forwarding of information. We have also introduced an entropy-based notion of privacy preservation, and analyzed the tradeoff between privacy preservation and forwarding accuracy by means of both analysis and simulation. The most important contribution of our study is demonstrating for the first time that privacy preservation and application-level performance must be traded off with each other,

and that the optimal tuning of this tradeoff depends on the specific application requirements.

We believe the study reported in this paper discloses several avenues for further research in the field. A first interesting direction is extending our study to other mobile social networking applications, such as small talking [10], Twitter [9], etc. More research is also needed to demonstrate feasibility of the secure multi-party computation framework proposed herein, as well as in [11], [12], [13], [14], on a mobile platform. Initial steps along this direction are reported in [27] .

ACKNOWLEDGMENT

Work partially supported by the EU projects FP7-257930 Aniketos and FP7-256980 Nessos. The work of P. Santi was partially supported by MIUR, program PRIN, Project COGENT.

REFERENCES

[1] in *http://en.wikipedia.org/wiki/Mobile_phone*, 2012.
[2] in *http://en.wikipedia.org/wiki/Broadband_penetration*, 2012.
[3] in *http://www.loopt.com/*, 2012.
[4] in *http://www.mobiluck.com/*, 2012.
[5] in *http://en.wikipedia.org/wiki/Nokia_Sensor*, 2012.
[6] S. Gaonkar, J. Li, R. R. Choudhury, L. Cox, and A. Shmidt, "Microblog: Sharing and querying content through mobile phones and social participation," in *ACM Mobisys*, 2008.
[7] K. Li, T. Sohn, S. Huang, and W. Griswold, "Peopletones: A system for the detection and notification of buddy proximity on mobile phones," in *ACM Mobisys*, 2008.
[8] M. Motani, V. Srinivasan, and P. Nuggehalli, "Peoplenet: Engineering a wireless virtual social network," in *AMC Mobicom*, 2005, pp. 243–257.
[9] N. Ristanovic, G. Theodorakopoulos, and J.-Y. LeBoudec, "Trap and pitfalls of using contact traces in performance studies of opportunistic networks," in *IEEE Infocom*, 2012, pp. 1377–1385.
[10] Z. Yang, B. Zhang, J. Dai, A. C. Champion, D. Xuan, and D. Li, "E-smalltalker: A distributed mobile system for social networking in physical proximity," in *IEEE ICDCS*, 2010, pp. 468–477.
[11] M. Li, N. Cao, S. Yu, and W. Lou, "Findu: Privacy-preserving personal profile matching in mobile social networks," in *IEEE Infocom*, 2011, pp. 2435–2443.
[12] E. Baglioni, L. Becchetti, L. Bergamini, U. Colesanti, L. Filipponi, A. Vitaletti, and G. Persiano, "A lightweight privacy-preserving sms-based recommendation system for mobile users," in *ACM RecSys*, 2010.
[13] W. Dong, V. Dave, L. Qiu, and Y. Zhang, "Secure friend discovery in mobile social networks," in *IEEE Infocom*, 2011, pp. 1647–1655.
[14] R. Zhang, Y. Zhang, J. Sun, and G. Yan, "Fine-grained private matching for proximity-based mobile social networking," in *IEEE Infocom*, 2012, pp. 1969–1977.
[15] G. Costantino, F. Martinelli, and P. Santi, "Privacy-preserving interest-casting in opportunistic networks," in *IEEE Wireless Communication and Networking Conference (WNCN)*, 2012.
[16] A. Mei, G. Morabito, P. Santi, and J. Stefa, "Social-aware stateless forwarding in pocket switched networks," in *IEEE Infocom*, 2011.
[17] S. Jain, K. Fall, and R. Patra, "Routing in delay tolerant networking," in *ACM Sigcomm*, 2004, pp. 145–158.
[18] C. Andrew and C. Yao, "Protocols for secure computations," in *23rd IEEE Symposium on FOCS*, 1982, pp. 160 –164.
[19] C. Boldrini, A. Passarella, and M. Conti, "Contentplace: Social-aware data dissemination in opportunistic networks," in *ACM MSWiM*, 2008, pp. 203–210.
[20] V. Lenders, M. May, G. Karlsson, and C. Wacha, "Wireless ad hoc podcasting," *Mobile Computing and Communications Review*, vol. 2, pp. 65–67, 2008.
[21] A.-K. Pietlainen, E. Oliver, J. LeBrun, G. Varghese, and C. Diot, "Mobiclique: Middleware for mobile social networking," in *ACM WOSN*, 2009, pp. 49–54.
[22] O. Goldreich, *Foundations of Cryptography*. Cambridge University Press, 2001, vol. Basic Tools.
[23] R. Baeza-Yates and B. Ribeiro-Neto, *Modern Information Retrieval*. New York: ACM Press - Addison-Wesley, 1999.
[24] M. J. Freedman, K. Nissim, and B. Pinkas, "Efficient Private Matching and Set Intersection," in *EUROCRYPT*, ser. LNCS, vol. 3027, 2004, pp. 1 – 19.
[25] S. Li, D. Wang, and Y. Dai, "Symmetric cryptographic protocols for extended millionaires' problem," *Information Sciences*, vol. 52, no. 6, pp. 974 – 982, 2009.
[26] I. Ioannidis and A. Grama, "An efficient protocol for Yao's millionaires' problem," *System Sciences, 2003. Proc. of the 36th Annual Hawaii Int. Conf.*, p. 6, 2003.
[27] G. Costantino, F. Martinelli, P. Santi, and D. Amoruso, "An implementation of secure two-party computation for smartphones with application to privacy-preserving interest-cast," in *International Conference on Privacy, Security and Trust (PST)*, 2012.
[28] D. Malkhi, N. Nisan, B. Pinkas, and Y. Sella, "Fairplay: a secure two-party computation system," in *Proc. USENIX Security Symposium*, Berkeley, CA, USA, 2004.
[29] C. Shannon, "A Mathematical Theory of Computation," *Bell Systems Technical Journal*, vol. 27, pp. 623–656, 1948.
[30] B. Köpf and D. A. Basin, "An information-theoretic model for adaptive side-channel attacks," in *ACM Conference on Computer and Communications Security*, 2007, pp. 286–296.
[31] G. Smith, "Quantifying information flow using min-entropy," *Quantitative Evaluation of Systems, International Conference on*, vol. 0, pp. 159–167, 2011.
[32] P. Flajolet, D. Gardy, and L. Thimonier, "Birthday Paradox, Coupon Collectors, Caching Algorithms and Self-Organizing Search," *Discrete Applied Mathematics*, vol. 39, no. 3, pp. 207–229, 1992.
[33] D. Newman and L. Shepp, "The Double Dixie Cup Problem," *American Mathematical Monthly*, vol. 67, pp. 58–61, 1992.
[34] N. Eagle and A. Pentland, in *CRAWDAD data set mit/reality*, 2005.

G. Costantino received his Ph.D. degree from the University of Catania (Italy) in 2011. This followed his Master and Bachelor degrees from the same University, respectively in 2007 and 2005. He currently is a post-doc researcher at the Institute of Telematics and Informatics (IIT) of the National Research Council (CNR) located in Pisa (Italy). He is involved in two main research projects: mechanisms of trust management for Web Services, and preservation of users' privacy within Opportunist Networks.

F. Martinelli received his PhD in Computer Science from the University of Siena in 1999. He is a senior researcher of IIT-CNR. He is co-author of more than 100 scientific papers, and his research interests range from formal methods, distributed systems, computer security and foundations of security and trust. He founded and chaired the WG on security and trust management (STM) of the European Research Consortium in Informatics and Mathematics (ERCIM), and he is involved in several Steering Committees of international WGs and/or Conferences/workshops.

P. Santi received the Ph.D. degree in computer science from the University of Pisa in 2000. He has been researcher at the Istituto di Informatica e Telematica del CNR in Pisa, Italy, since 2001. His current research interests include the investigation of fundamental properties of wireless multihop networks. He has contributed more than 70 papers and two books in the field of wireless ad hoc and sensor networking, he is/has been AE of IEEE Trans. on Mobile Computing and IEEE Trans. on Parallel and Distributed Systems. He is a member of IEEE Computer Society and a Senior member of ACM.