A framework for privacy analysis of ICN architectures

Nikos Fotiou[1], Somaya Arianfar[2] Mikko Särelä[2], and George C. Polyzos[1]

[1] Mobile Multimedia Laboratory, Athens University of Economics and Business, Athens, Greece
{fotiou,polyzos}@aueb.gr,
[2] Department of Communication and Networking, Aalto University, Helsinki, Finland {somaya.arianfar,mikko.sarela}@aalto.fi,

Abstract. Information-Centric Networking (ICN) has recently received increasing attention from the research community. Its intriguing properties, including identity/location split, in-network caching and multicast, have turned it into the primary paradigm for many recent internetworking proposals. Most of these are mainly concerned with core architectural issues of ICN including naming, routing, and scalability, giving little or no attention to *privacy*. Privacy issues however, are together with security, an integral part of any contemporary communication technology and play a crucial role for its adoption. Since the core functions of an ICN architecture are content name based, many opportunities for privacy related attacks–such as user profiling–are created; being aware of these privacy threats, users might completely dismiss the idea of using an ICN-based network infrastructure. Therefore, it is important to investigate privacy as an integral part of any ICN proposal. To this end, in this paper, we develop a privacy framework for analyzing privacy issues in different ICN architectures. Our framework defines a generic ICN model as well as various design choices that can be used in order to implement the functions of this model. Moreover it considers a comprehensive list of privacy attack categories, as well as various types of adversaries.

1 Introduction

Information-Centric Networking (ICN) is an emerging paradigm that has received increasing attention in recent years. ICN is believed to overcome various limitations of the current networking architectures, including inefficient mobility handling, lack of effective multicast, insecurity and distorted business environment. A key property of ICN architectures[3] is the use of content names as a new abstraction layer between applications and the network. In contrast to the IP model that relies on endpoints location and IP addresses, ICN relies on content names to provide the expected networking functionality. Communicating entities in ICN architectures reveal the name of the content to the network, either to make it available to others or to ask for the network to retrieve it. Using this information,

[3] See [15] for a survey on ICN research efforts

it is believed, that the network can provide better services to networked applications and to interconnect different application domains [14].
This change in the communication model also changes the privacy model of the network [8]. Today, the network only sees the IP-address of entities that communicate with each other. A secure encrypted channel can be established in order to prevent the network from seeing what is actually being transmitted between endpoints. However, in an ICN based architecture, where users access the network using content names, the network should be able to recognize this information and use it in various networking functions. In addition to exposing the content name to the network, various forms of privacy threats can be created depending on the specific design and implementation choices.
Discussing these new forms of privacy threats, before anything else, requires the understanding of an ICN architecture and this includes the breaking down of the architecture into core components and the identification of the ways these components interact with each other. Since the ecosystem of ICN is composed by a significant number of heterogeneous architectures, defining a common model that captures all of them is not a trivial task. A significant part of this work is devoted to creating such a common and proposal-independent model of ICN design. This model identifies roles and functions that are common in many ICN proposals and presents various design choices for implementing these functions.
Understanding and modeling an ICN architecture is the first step towards its privacy analysis. However in order to reach this target another step is required: the identification and documentation of the privacy threats. To this end, we present a thorough list of privacy attacks categories, we define various adversary types and we use an existing methodology for documenting threats.
The reminder of the paper is organized as follows. We discuss related work in Section 2. We define a generic model of ICN and we present available design choices in Section 3. In Section 4, we discuss types of privacy attacks and adversaries and we use the DREAD [9] methodology to analyze privacy threats. In Section 5 we present various research efforts in the area of ICN privacy. Finally conclude our paper in Section 6

2 Related work

To our knowledge privacy in ICN has only been discussed in a few other works, mainly by Lauinger et al.[11] and by Chaabane et al.[3]. Lauinger et al.[11] identify three privacy threats: *information leakage through caches*, *censorship* and *surveillance*. In the first type of attacks a malicious entity tries to learn which users are interested in a content item by requesting this item and by measuring the response time: small response times are an indication that the requested item has been cached close to the malicious entity. If caches are used at lower aggregation levels then the number of users that share a cache will be limited, therefore it might be possible to associate cached content items with the users that originally requested it. Moreover, censorship and surveillance attacks are possible since content items in ICN are uniquely identified, therefore it

is easier for a privileged malicious entity to either block specific items or monitor the users that access specific items.
Chaabane et al.[3] identify four categories of privacy attacks related to: *caching*, *content*, *naming* and *signatures*. Similar to [11], cache privacy attacks exploit response times and–potentially–reveal the preferences of a group of users. Chaabane et al. distinguish the adversaries with respect to this kind of attack, into *immediate* neighbors and *distant* neighbors, with immediate and distant referring to the network distance between the adversary and the target. Content privacy attacks aim at monitoring and censoring users and they are facilitated by the fact that content is cached, therefore an attacker has more time to inspect the data. Name based privacy attacks are enabled due to the semantic correlation between the content and its name. These attacks are amplified by the fact that content name cannot be easily encrypted, because it is needed by the networking functions. Finally signature privacy attacks refer to attacks that target a content owner that has digitally signed content data in order to protect its *provenance*.
Both works by Lauinger et al.[11] and by Chaabane et al.[3] assume certain design choices, inspired by the NDN ICN architecture [10]. Although the same privacy threats may exist under different setups, their impact and their method of exploitation varies. Our work is not limited to particular design choices. On the contrary, we propose a generic model for ICN, we discuss various design choices and we argue how these design choices affect the feasibility and the impact of privacy attacks. The attacks described in these works are captured by our model and they are discussed in more detail. Moreover our model defines additional privacy threats and proposes a richer adversary model. Finally these works propose solutions for these attacks, based on the NDN ICN architecture. The goal of our paper is not to propose a specific security solution: its goal is to set the foundations of a privacy framework that will allow the assessment of a privacy risk and the measurement of the effectiveness of a privacy solution.
Any generic privacy analysis framework (e.g., [4]) can be used (with small or big modifications) for the privacy analysis of an ICN architecture. However, we believe that a framework tailored for this paradigm can accelerate this process and facilitate the detection of new privacy threats and of critical design choices.

3 An ICN model

In this section we identify the main roles and functions that may exist in an ICN network. We then dicuss different design choices available to support the expected functionalities.

3.1 Roles and functions in ICN

Generally an ICN architecture is composed of the following entities[4]

[4] The terminology is not entirely standard because various architectures, designs and research efforts more generally, have different priorities

- (Content or information) Owner: The entity that creates and owns a content item. The owner is responsible for assigning names to content items and for creating (if necessary) access control rules that govern who can access each item. The role of owner captures real world entities (e.g., an author, a university, a company, a government)
- Consumer: The entity that is interested in receiving (access) a content item. A consumer is a real world entity that interacts with the network through a device (e.g., a computer, a mobile phone). In the rest of the paper when stated that a consumer interacts with the network, it is always meant through his access device.
- Storage node: A network entity that actually hosts a content item. A storage node may be under the full control of an owner (e.g., the web server of a university), but it may also be (semi-)independent (e.g., proxy caches and CDN servers). Storage nodes may either have been appointed by the owners themselves (e.g., a university may host a content item in its web server, or pay a CDN to host it), or may act opportunistically (e.g., an in-network cache).
- Resolver: A network entity that acts as an indirection point between consumers' devices and storage nodes. A resolver's main functionality is to accommodate consumers' interests for particular content items. All the resolvers of an architecture form the *resolution network*.

These entities interact with each other in the following manner: An owner creates a content item, assigns a *name* to it and stores a copy of this item in at least one storage node. The storage nodes *advertise* the content items they host. The advertisement of an item is received and kept by some resolvers in the network. A consumer sends a content *lookup* request that is routed through the resolution network and eventually reaches a resolver that has a matching entry for that item of interest. A successful match will ultimately result in the content being *forwarded* from a storage node to the interested consumer(s). Intermediate nodes may opportunistically *cache* a forwarded item, and act as additional storage nodes for that item in the future.

3.2 Design choices for content naming

The choice of a naming structure for an ICN, depends on various properties expected from each naming scheme. Some basic properties include:

Security bindings In ICN, the network has to ensure the *authenticity* of the content items. Therefore, the network–or some specific entity in the network–has to make sure that a name is associated with the correct content item. This requires either a direct or an in-direct binding between the content and its name. With a direct binding, the name or a part of the name is cryptographically derived from the content. With an in-direct binding, the name is securely bound to an entity which can vouch for the rightfulness of the link between a content item and a name.

Human readability Human readable names can be easily memorized by users. Usually they are of varying length, and because they are

meaningful and distinguishable to the users some names become more popular than others. Thus, in some cases using human readable names require the existence of a naming assignment authority that handles various issues, such as, copyrights. Names that are not human readable, are usually of constant size and indistinguishable by users. Non-human readable names can be derived by using mechanisms such as (secure) hash functions. With these names usually a search engine-like mechanism is required, in order to map a human readable description to a content name.

Mutability A content name can be mutable or immutable. Mutable names are short lived. When mutable names are used, mechanisms for finding the current name of a content item and for examining if this name is still valid, should be considered. Immutable names are long lived. When immutable names are used, entities that assign names may be required, otherwise the architecture may suffer from conflicts among owners who wish to use the same name for their different content.

Content to name mapping The final property of a naming scheme concerns how many names can a content item have simultaneously. A content item may have multiple names or a single one. When an item has multiple names it may (or may not) be possible to tell if two names identify the same object or not.

3.3 Design choices for advertisement and lookup

Content advertisement creates state in the resolution network that is used for routing content lookup messages to a storage node that hosts the desired item. The routing of the advertisement and lookup messages may be logically *coupled* or *decoupled* to the routing protocol of the architecture.

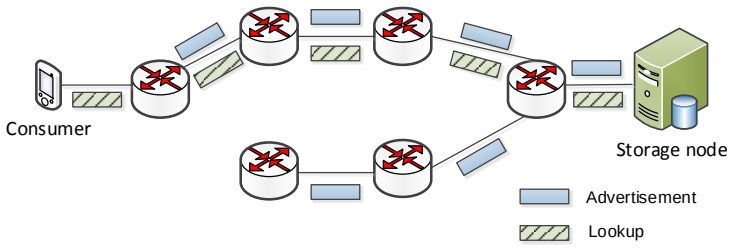

Fig. 1. Advertisement and lookup coupled to the routing protocol. All network routers act as resolvers

When advertisement and lookup are coupled to the routing protocol, the corresponding messages directly shape (and follow) the routing table entries of all routers all over the network. Content advertisements are flooded to the whole network and create the routing state. Lookup messages are routed using this state to an appropriate storage node. When this design choice is used, the resolution network is formed by all network routers. Figure 1 gives an example of an ICN network, where advertisement and lookup are coupled to the routing protocol.
When advertisement and lookup are decoupled to the routing protocol the resolution network is implemented as a new overlay network. Content advertisements are routed in the overlay network until they reach a specific resolver that is responsible for handling the advertised content name; this resolver acts as the *rendezvous point* for this content name. The advertisement messages create state only in the rendezvous points. Content lookup messages are routed in the resolution network until they reach an appropriate rendezvous point; when a content lookup reaches the rendezvous point, the latter *notifies* a storage node. Figure 2 gives an example of an ICN network, where advertisement and lookup are decoupled from routing protocol.

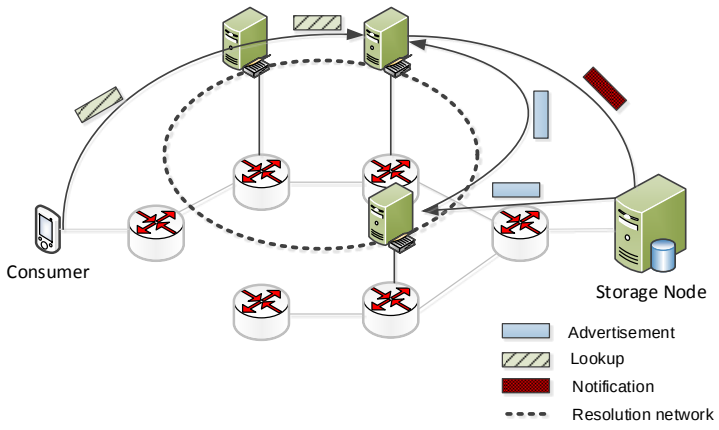

Fig. 2. Advertisement and lookup decoupled to the routing protocol. Resolvers are seperated entities organized in an overlay network

3.4 Design choices for forwarding

A successful advertisement/lookup match leads to the desired content item being forwarded to the consumer. An ICN architecture can be

geared towards using source-based forwarding or towards using hop-by-hop forwarding. In the former case, a storage node "learns" the path towards the consumer(s), and encodes it in a format that can be used by the intermediate nodes in order to take the appropriate forwarding decisions. The forwarded content items therefore should include this encoding in the header. When hop-by-hop forwarding is used data items are forwarded back to the consumer using the state that has been created during the lookup process: in this case every node that routes a lookup message maintains state that indicates the direction towards which the corresponding response should be forwarded. In this case, the forwarded content items should include in their header the same identifier as the one used during the lookup phase, and they should follow the same path as the lookup message.

4 Privacy threat analysis

In this section we present our privacy threat model. In this model two information containers are of importance *data flows* and *data pools*. A data flow concerns the data traversing the network in order to reach some endpoint(s) whereas a data pool concerns the data stored in a single node, usually used for facilitating networking operations and applications. Examples of data flows are advertisement messages, lookup messages, notifications, and forwarded items. Example of data pools is the state created in a resolver by the advertisement messages.

4.1 Adversaries

Adversaries may act on their own or collude with other entities. Adversaries in our framework are grouped by their *location*, *role*, and *mode of operation*.

With respect to the location an adversary can be *arbitrary* or *local*. An arbitrary adversary launches a privacy attack from an arbitrary point in the network. On the other hand a local adversary is located "close" to the target in terms of physical, network, or even social proximity. A malicious mobile phone in the same room as the target's laptop, a malicious default gateway, and a malicious consumer sharing the same interests with the target, all are examples of local adversaries.

An adversary can hold one or more of the following roles: *owner*, *consumer*, *storage node*, *resolver*, *observer*, or *authority*. The first four roles concern ICN roles and refer to ICN entities acting maliciously. An adversary holding the role of an observer is a third party that cannot actively participate in the defined procedures but has access to the data flows and data pools. An eavesdropper listening to the communication between a consumer and a resolver is an example of an adversary that has the role of the observer. An adversary that holds the role of the authority reflects an entity that either can administrate network elements of the architecture, or it is in position to dictate to some network elements how to behave. A resolver provider and a state government are two examples of adversaries that hold the role of the authority.

Finally adversaries may be *active*, *passive* or *honest-but-curious*. Active adversaries may change an information flow and/or a record in a data pool or completely remove it. Depending on the architecture and the particular implementation choices the actions of an active adversary may be *detectable* or *undectable*. As an example, if digital signatures are used in every information flow the manipulation of a data flow will be detectable with high probability. Passive and honest-but-curious adversaries simply observe data flows, and/or data pools, and/or *side channel* information (such as response times). The difference between a passive and honest-but-curious adversary is that the latter does not deviate from the specified protocols, whereas the former may violate them in order to achieve her goal (e.g., she may *impersonate* another entity).
How effective an adversary is into launching a privacy attack is highly affected by the number of the data flows and pools to which he has access, as well as, by the amount of information that is revealed by these flows and pools. Clearly this depends on the characteristics of the adversary, as well as, on the design choices that have been made.

4.2 Privacy attacks

The terminology of the attacks considered in our framework is borrowed from Solove [13]. However, the taxonomy of the attacks has been modified (compared to [13]) and has been adapted to the context of ICN.
Privacy attacks can be grouped into *monitoring* attacks, *decisional interference* attacks and *invasion* attacks. An attack may belong to multiple groups.

Monitoring Monitoring attacks aim at learning the preferences and interests of particular consumers, or the consumers interested in a particular content item (or group of items), or the types of content a particular owner offers, or the owners of a particular content item (or group of items). This goal can be achieved using the *surveillance* and *interrogation* attacks, the *identification* attack, and the *breach of confidentiality* and *disclosure* attacks.
Surveillance aims at collecting as much information about a target as possible. This information includes lookup messages, advertisements, forwarded items, as well as, side-channel information. Surveillance can be performed by passive or honest-but curious attackers (of any role) simply by monitoring data flows and pools or by active attackers that probe data pools (e.g., by requesting content from caches) or insert new data flows (e.g., repeat a lookup message). A surveillance attack can be supported (or amplified) by the interrogation attack. Interrogation aims at forcing targets into giving information in order to receive or take part in a service. Interrogation can be achieved for example by a resolver that requires owners to digitally sign their advertisements in order to be accepted. Malicious resolvers can potentially collect information using interrogation from specific owners and consumers, whereas malicious owners can potentially collect information using interrogation from specific

consumers. An interrogation attack is more effective when the adversary behaves in an honest-but-curious manner.
Identification aims at linking collected information to a particular target. An identification attack aims at linking: a data flow to a consumer or to an owner, or data flows to each other (e.g., a lookup to a response). An identification attack can be launched by any attacker capable of collecting information.
Breach of confidentiality and disclosure are both related to the revelation of information regarding a target by a third party. Breach of confidentiality refers to the revelation of information about a target, stored in a (previously) trusted entity. The impact of this attack is dual: it reveals information about the target and it breaks a trust relationship. Revealing a list of consumers of a content item by a resolver constitutes such an attack. The entity that reveals this information may be the trusted entity itself, or another third party. A breach of confidentiality attack can be performed by an active attacker that interacts with the target trusted entity, or by any attacker that holds an ICN entity role.
Disclosure occurs when certain information about a target is revealed to others. The revealed information is in transit or stored in an untrusted entity (e.g., a cache). Therefore a disclosure attack does not involves the break of a trust relationship. An example of disclosure attack is the revelation that a consumer is interested in a particular content item, by an attacker that monitors the the communication channel between a consumer and a resolver. Disclosure attacks can be performed by an active or passive observer.

Decisional interference Decisional interference attacks may aim at one or more of the following: (i) preventing a particular consumer from accessing certain content items, (ii) preventing the advertisement or forwarding of content items belonging to certain owners, (iii) preventing the advertisement or forwarding of content items that have certain characteristics (e.g., censorship based on content identifiers or filtering based on file types). This goal can be achieved using the identification attack followed by the *insecurity* or the *distortion* attack.
The insecurity attack refers to the manipulation of a data pool that is possible due to the inefficiencies or vulnerabilities of the way it is maintained. In other words the insecurity attack exploits the fact that a data pool is not properly secured and therefore illegitimate information can be added, or legitimate information can be removed. An example of this attack is the manipulation of the state of a resolver in order to erase advertisements of certain content items. Insecurity attacks can be performed by active attackers that exploit weaknesses in the implemented protocols.
Distortion, on the other hand, aims at manipulating or deleting an information flow in order to hide a consumers lookup, or an advertisement or a forwarded content item. Therefore this attack "distorts" the profile of a consumer and presents her as she is not interested in a content item, or "distorts" the profile of a storage node and presents it as it does not "serve" an item. Any active attacker can launch this type of attacks.

Invasion Invasion attacks affect privacy related information of a target in order to cause (not necessarily privacy related) harassment. In particular they aim at luring a consumer into requesting particular content items, force the forwarding of a content item to a consumer (not necessarily interested in that item), make a resolver associate a content item with a particular owner or storage node. Invasion is possibly using the *insecurity* and the *distortion* attacks, described previously, as well as using the *exclusion* and the *secondary use* attacks.

The insecurity attack is used in order to make a resolver believe that a consumer is interested in a particular content item, or that a storage node offers an item, or that an item belongs to a owner. The distortion attack is used in order to modify a lookup, or an advertisement or a forwarded item in order to refer to another content item, or consumer, or storage node, or owner.

Exclusion prevents a target from modifying or deleting an entry stored for him in a data pool. As an example if a consumer is not able to withdraw his interest on a specific content item, this may result in receiving items in which he is not interested in. Malicious resolvers maintain information about both consumers and owners therefore they may prevent them from modifying it. Similarly malicious owners may maintain information about consumers. Finally, active attackers, may *block* messages that aim at modifying stored information.

Secondary use, is the use of collected information for purposes unrelated to the purposes for which the information was initially collected without the target's consent. An example of this attack is the repetition of a lookup message. Any adversary that is able to collect information can potentially perform this attack.

Figure 3 illustrates the identified privacy threats.

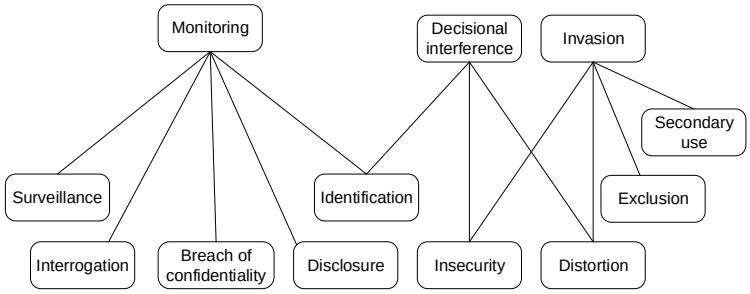

Fig. 3. ICN Privacy threats

4.3 Analyzing threats

The ultimate goal of a privacy analysis is to identify and document privacy threats. The prerequisites for this step are: a model of the ICN architecture that specifies the design choices, a list of considered privacy attacks and a list of adversary types. Given this information threats can be ranked based on their feasibility and impact. The DREAD model [9] can be used to achieve this goal. DREAD is a threat ranking model, developed by Microsoft, that ranks threats based on their **D**amage, **R**eproducibility, **E**xploitability, **A**ffected users and **D**iscoverability.
Before proceeding with an example, we revisit the design choices, presented in Section 3, and for each choice we identify some properties that may affect a DREAD factor. This list is not exhaustive and does not consider combinations of choices.

Naming	
Direct security binding	Content items can be tracked more easily
Indirect security binding	Additional entities are required (therefore more potential adversaries)
Human readable	Names reveal more information
Human unreadable	Additional entities may be required
Mutable	Additional entities may be required
Immutable	Content items can be tracked more easily
Single name per item	Content items can be tracked more easily
Multiple names per item	Additional entities may be required
Advertisement, Lookup	
Coupled	Lookups and forwarded content may traverse the same nodes, any network router can be a potential resolver for a specific item
Decoupled	Resolvers have greater power, lookups should contain a consumer specific location identifier
Forwarding	
Source-based	Forwarded items contain information about how can a consumer be reached
Hop-by-hop	Forwarded items usually contain the content identifier

Table 1. Privacy related properties of the design choices

Let us now illustrate how DREAD model can be used in our framework using an example.

For each DREAD factor we use a scale from 1 to 5. Our setup is the following

- `Privacy threat`: Surveillance of the consumers of a specific content item
- `Design choices`: Advertisements and lookups are decoupled to the routing protocol, names are immutable, each content item is identified by a single name
- `Adversaries`: Arbitrary, honest-but-curious resolver

In this setup all lookups for a specific content identifier will end up in the same resolver. Since the content item is identified by a single name there will be a resolver for handling all requests for this content item. If the adversary happens to be that resolver then it is able to monitor *all* lookups for that content item. The `Damage` factor of this threat will receive therefore the highest rank (5). On the other hand, generally, it is not very easy for an attacker to make the resolution network to believe she is responsible of a particular content name (of course this is implementation specific, but we base this assumption considering how DNS and secure DHTs are organized), therefore the `Reproducibility` of the attack will not receive a high rank (1). In order for a malicious resolver to perform this attack it has simply to observe incoming lookups. Moreover lookups should contain a location identifier of the consumer therefore they can be relatively easily linked the to particular consumers. So `Exploitability` will also receive a high rank (4). This attack may potentially affect all the consumers of the content item. The number of these consumers depends on the popularity the item. Therefore `Affected users` will receive an indicative rank (3). Since this is a passive attack, it cannot be easily discovered. However if it is discovered, it is easy to decide which resolver performed it. Therefore, `Discoverability` will receive a medium rank[5] (2). The following table summarizes our assessment.

Damage	Reproducibility	Exploitability	Affected users	Discoverability
5	1	4	3	2

Table 2. DREAD ranking

Let's now modify the design choices of the ICN architecture and examine how the ranking is of this privacy threat is affected. We now consider a setup with the same threat and adversary model but with the following design choices:

- `Design choices`: Advertisements and lookups are **coupled** to the routing protocol, names are immutable, each content item is identified by a single name

[5] The Discoverability rank indicates how hard is to detect a threat: the higher the rank the harder is to detect a threat

When advertisements and lookups are coupled, lookups use the routing plane. Therefore, in contrast to the previous scenario it is not likely that all lookups for a specific content will be routed through the same resolver. The `Damage` factor of this threat will receive a lower rank (2). On the other hand, compared to the previous setup, it is easier for a resolver to make the resolution network to believe that it knows a route to a particular content name therefore the `Reproducibility` of the attack will receive higher rank (3). As in the previous setup, in order for a malicious resolver to perform this attack it has simply to observe incoming lookups. However in this setup lookups contain only the next hop to the consumer, so additional information is required in order to link a lookup to a consumer. Therefore `Exploitability` will receive a lower rank (3). The design choices do not affect the popularity of the item, therefore `Affected users` will receive the same indicative rank (3) . Finally, again this attack cannot be easily discovered. However even if it is discovered, it is not easy to detect which resolver performed it. Therefore, `Discoverability` will receive higher rank (3). Table 3 summarizes the assessment for this setup.

Damage	Reproducibility	Exploitability	Affected users	Discoverability
2	3	3	3	3

Table 3. DREAD ranking

Although subjective, the DREAD ranking gives an indication how the design choices affect the privacy properties of an ICN architecture. In the studied case, if it is assumed that all DREAD factors are equally wighted it can be concluded that the second design choice has better privacy properties w.r.t. to the specific threat model.

5 ICN Privacy research

Various research efforts have highlighted privacy issues in ICN architectures and they have proposed solutions to address them.
DiBenedetto et al.[5] proposed a Tor-like *anonymization* network for the NDN ICN architecture [10] code-named ANDaNA. In ANDaNA, before sending a lookup request, a consumer selects two "anonymizing routers", the entry router and the exit router, and distributes different symmetric encryption keys to each of them. The consumer encrypts her request using the public keys of the routers, and sends the request to the entry router, which then forwards it to the exit router. When the exit router receives a response, it encrypts it using the symmetric key has been provided by the consumer, and forwards it to the entry router. Then the entry router encrypts once more the received ciphertext with its own symmetric key (that has been provided by the consumer) and forwards the response to the consumer. Finally, the consumer decrypts

the response. ANDaNa protects consumers against *surveillance* and *distortion* attacks, since their lookups and the corresponding responses are encrypted and their integrity is checked (although additional measures are required in order to detect deleted lookups or forwarded items). The proposed scheme offers protection against malicious storage nodes and observers. A malicious resolver that happens to be the entry router learns the identity of the consumer and the identifier of the item in which she is interested in, whereas a malicious resolver which happens to be the exit router learns the content item identifier and potentially its data. The former resolver is able to perform distortion and possible surveillance, whereas the latter resolver is able to perform decisional interference.

Arianfar et al. [2] proposed a solution that offers *pseudonymity* of content names for the PURSUIT [6] architecture. In their approach an owner splits the file she wants to protect into n blocks, t_1, t_2,..., t_n, and creates a "cover file" with n blocks (c_1, c_2, ..., c_n). All file blocks, and the corresponding cover file blocks, are assumed to have the same length. Then, the owner applies a reversible randomizing function r() to every block and advertises all the (randomized) blocks of the cover file (i.e., $r(c_1)$, $r(c_2)...r(c_n)$) as well as chunks that are created by XORing a (randomized) file block with a (randomized) covered file block (e.g., $r(t_1)$ XOR $r(c_2)$, $r(t_3)$ XOR $r(c_1)$). For a consumer to receive a file block she has to lookup for to the appropriate cover file blocks and chunks (e.g., in order to receive t_1 she must perform a lookup for to $r(c_2)$ and to $r(t_1)$ XOR $r(c_2)$, and then she will be able to compose t_1, simply by XORing the received packets). The name used for the i^{th} advertised block of the cover file c is $H(H(c), i)$, where H is a well known function. The name used for an advertised chunk, composed by XORing the k^{th} block of the cover file with the l^{th} file block of a file t is $H(H(c), k, H(t), l)$. A consumer learns through a secured channel the function $H()$ and the number of blocks, therefore she is able to perform lookups for any combination of files. An attacker on the other hand, is not able to determine the file that in which a consumer is interested in. Providing that the cover file is updated often enough, the proposed solution protects consumers and owners from *surveillance* by malicious observers, since the content identifiers used for both advertisement and lookup are scrambled. Moreover, this solution offers protection against *insecurity* and *distortion* attacks, targeting specific content item identifiers. However a malicious resolver is able to determine the owner with which a consumer interacts and vice versa.

Fotiou et al. [7] proposed a solution that offers *unobservability* of data flows for the PURSUIT architecture. The unobservability property assures that it is not possible to associate a data flow with a particular content item. The proposed solution is based on the homomorphism property of the Paillier cryptosystem [12], which allows operations over encrypted data by a 3^{rd} party without revealing to that 3^{rd} party any information associated with this data. The approach is based on a query/response model in which a consumer defines a linear equation over a set of content item identifiers and a resolver solves this equation. The result of this equation is the location identifier of the item in which the consumer is really interested in. Nevertheless, the resolver is unable to interpret the

result as it is encrypted with a key that is known only to the consumer. The solution completely hides consumer preferences from observers and resolvers, therefore, it protects consumers from *surveillance*. Moreover the proposed scheme performs integrity checks and prevents lookup repetitions, thus protecting consumers from *distortion* and *secondary use* attacks.

Many recent works, study the problem of consumer *surveillance* by malicious observers using as a side-channel information the response time of a content lookup. Lauinger et al.[11] as well as Acs et al.[1] assess this problem in the context of NDN. In NDN a local cache in an access network is often populated with the items accessed by a few users in its vicinity. In this case if an adversary can figure out which items have been cached, it can easily associate those items with a certain group of local consumers. Chaabane et al.[3] point out that specific protocol details can increase the chances of cache tracing in NDN. Specifically, NDN's prefix-based content request and delivery means that an adversary can just ask for a certain prefix and the cache would return any available item with that prefix. There are different solutions suggested in [11, 1, 3] to overcome the problem of tracing the cache access pattern. These solutions can be divided into two different categories: first, affecting the access pattern or the cache structure, and second, changing the content or its name and affecting the cacheability of each item. Access patterns can be obfuscated for privacy considerations, e.g. through adding random delay to the data that is served from a cache, or through caching only the items that have been accessed at least K times. The cache structure can be affected using collaborative caching and by increasing the size of the user-set for each cache. The cachebility of private items can be affected by creating new, user-specific, names or by flagging the content item as being private and not cacheable.

6 Conclusions

ICN is an intriguing networking paradigm receiving growing attention. However, being name oriented, ICN raises privacy concerns, which unfortunately have not been tackled by the research community. Privacy analysis of ICN is impeded by the lack of a privacy analysis framework, which is mainly due to the departure from the traditional end-host oriented communication model, as well as to the multitude of different ICN architectures.

In this paper we developed a generic, solution independent, model of an ICN architecture and we highlighted the design choices that can be made for implementing its functions. We believe that most ICN architectures can be mapped onto this model. Moreover, we presented a thorough list of categories of privacy attacks, as well as a comprehensive adversary model. These tools can be used for identifying and ranking privacy risks in existing and future ICN architectures.

Future work in this domain includes the expansion of our model in order to include even more design choices, threats and adversary types, as well as the application of our model to evaluate the privacy properties of specific ICN proposals.

Acknowledgment

This research was supported by a grant from the Greek General Secretariat for Research and Technology, financially managed by the Research Center of AUEB.

References

1. Acs, G., Conti, M., Gasti, P., Ghali, C., Tsudik, G.: Cache privacy in named-data networking. In: the 33rd International Conference on Distributed Computing Systems (ICDCS) (2013)
2. Arianfar, S., Koponen, T., Raghavan, B., Shenker, S.: On preserving privacy in content-oriented networks. In: In Proc. ACM SIGCOMM workshop on Information-centric networking (ICN), pp. 19–24 (2011)
3. Chaabane, A., De Cristofaro, E., Kaafar, M.A., Uzun, E.: Privacy in content-oriented networking: threats and countermeasures. SIGCOMM Comput. Commun. Rev. **43**(3), 25–33 (2013)
4. Deng, M., Wuyts, K., Scandariato, R., Preneel, B., Joosen, W.: A privacy threat analysis framework: supporting the elicitation and fulfillment of privacy requirements. Requirements Engineering **16**(1), 3–32 (2011). DOI 10.1007/s00766-010-0115-7. URL http://dx.doi.org/10.1007/s00766-010-0115-7
5. DiBenedetto, S., Gasti, P., Tsudik, G., Uzun, E.: Andana: Anonymous named data networking application (2012)
6. Fotiou, N., Nikander, P., Trossen, D., Polyzos, G.C.: Developing information networking further: From psirp to pursuit. In: I. Tomkos, C. Bouras, G. Ellinas, P. Demestichas, P. Sinha (eds.) Broadband Communications, Networks, and Systems, *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, vol. 66, pp. 1–13. Springer Berlin Heidelberg (2012)
7. Fotiou, N., Trossen, D., Marias, G.F., Kostopoulos, A., Polyzos, G.C.: Enhancing information lookup privacy through homomorphic encryption. Security and Communication Networks (2013)
8. Ghodsi, A., Shenker, S., Koponen, T., Singla, A., Raghavan, B., Wilcox, J.: Information-centric networking: seeing the forest for the trees. In: Proceedings of the 10th ACM Workshop on Hot Topics in Networks, HotNets '11, pp. 1:1–1:6. ACM, New York, NY, USA (2011). DOI 10.1145/2070562.2070563. URL http://doi.acm.org/10.1145/2070562.2070563
9. Howard, M., LeBlanc, D.: Writing Secure Code 2nd Edition. Microsft Press (2002)
10. Jacobson, V., Smetters, D.K., Thornton, J.D., Plass, M.F., Briggs, N.H., Braynard, R.L.: Networking Named Content. In: Proc. ACM CoNEXT (2009)
11. Lauinger, T., Laoutaris, N., Rodriguez, P., Strufe, T., Biersack, E., Kirda, E.: Privacy risks in named data networking: what is the cost of performance? SIGCOMM Comput. Commun. Rev. **42**(5), 54–57 (2012)

12. Paillier, P.: Public-key cryptosystems based on composite degree residuosity classes. In: J. Stern (ed.) Advances in Cryptology- EUROCRYPT 99, *Lecture Notes in Computer Science*, vol. 1592, pp. 223–238. Springer Berlin Heidelberg (1999)
13. Solove, D.J.: A taxonomy of privacy. University of Pennsylvania Law Review pp. 477–564 (2006)
14. Trossen, D., Särelä, M., Sollins, K.: Arguments for an information-centric internetworking architecture. SIGCOMM Computer Communication Review **40**(2), 26–33 (2010)
15. Xylomenos, G., Ververidis, C., Siris, V., Fotiou, N., Tsilopoulos, C., Vasilakos, X., Katsaros, K., Polyzos, G.C.: A survey of information-centric networking research. Communications Surveys Tutorials, IEEE **PP**(99), 1–26 (2013)