To Route or To Secure: Tradeoffs in ICNs over MANETs

Hasanat Kazmi* Hasnain Lakhani Ashish Gehani
SRI

Rashid Tahir
University of Illinois, Urbana-Champaign

Fareed Zaffar
Lahore University of Management Sciences

***Abstract*—Information-Centric Networks (ICNs) operating over Mobile Ad hoc Networks (MANETs) are challenged by the node churn, evolving topologies, and limited resources of the underlying network. The complex interplay of publishers, subscribers, and brokers brings with it a corresponding set of security concerns, where precisely-defined trust boundaries are needed to guarantee the confidentiality and integrity of all data objects in the ecosystem. Building a practical framework that can service users efficiently requires understanding the motivations and actions of the participants.**

We explore several tradeoffs between efficiency and the security of data objects in such environments, using ICEMAN – a real-wold implementation of an ICN that operates on MANETs. Since our findings are based on an actual system, they have significant implications for building efficient ICNs that have security designed in at the outset (rather than added later when options may be limited). We empirically establish that there is a strong interplay between the need to have more specific information for efficient routing and the need to ensure trust and confidentiality in such a decentralized system.

I. Introduction

Information-Centric Networking (ICN) is an approach for content distribution and retrieval that has drawn considerable attention in recent years. While there are several competing architectures and implementations, the underlying idea is that data is de-coupled from a single location. Network functionality is driven by descriptions of content rather than requests for the content at a specific address, as occurs in traditional source-destination based models. Publishers can advertise descriptions of their content. Subscribers advertise their interests in the hope that they will flow to others that have relevant objects. Data transport and routing decisions are content-aware, driven by matches between interests of nodes and the descriptions of the published content. The network can therefore take advantage of various performance optimizations, such as in-network caching in order to reduce latency and improve bandwidth utilization. However, these useful characteristics of ICNs come with a corresponding set of trust, privacy, and security challenges that need to be addressed.

Mobile Ad hoc Networks (MANETs) are used in environments where nodes can join or leave the network at will. In such high-churn environments, the ability to authenticate nodes in the absence of a single online trusted authority becomes fundamental for the correct and secure operation of the network. In the absence of an authentication scheme, a malicious node can access private objects and generate spurious content to overwhelm the network using resource exhaustion attacks. Similarly, the confidentiality of metadata is an important concern as its breach can lead to privacy compromises. In particular, query, response, and forwarding information can reveal sensitive details about publishers, subscribers, and content [2]. For instance, descriptive information about a data object is usually embedded in the associated metadata. ICN routing algorithms leverage these content descriptions to match objects with subscriber interests for forwarding decisions. However, the information present in the metadata also leaks privacy-sensitive details about the nodes involved in the production and consumption of the data objects.

Several approaches have been suggested in the literature to improve the security, privacy, and confidentiality of ICN-based publish-subscribe systems [5], [4], [11], [12], [1]. However, enhancing security and privacy generally leads to a suboptimal data delivery model with degraded performance. For instance, reducing the number of forwarding options at routers (to enhance security) causes data objects to follow longer paths. This results in reduced data rates and higher network congestion [14]. Additionally, this can also lead to routing *black holes* for data objects – that is, a data object might never be able to find a path to the subscriber [9]. Hence, a detailed investigation is needed to better understand the tradeoffs between the efficiency of the system and the security, privacy, and confidentiality of the actors and data involved. To this end, we present a detailed analysis and results from our measurement- and modeling-based study of security in ICEMAN, SRI International's open source implementation of an ICN for MANETs [15]. Specifically, we explore three tradeoffs: (i) the effect on routing performance when nodes must be authenticated, (ii) how caching policies affect access control performance (since in-network communication is used to retrieve credentials), and (iii) the impact on content delivery rates when stronger privacy protections are utilized for data descriptions and subscriber interests.

The rest of the paper is organized as follows. Section II describes ICEMAN and its privacy-enhancing architecture. Section III explains the tradeoffs we will examine. Our experimental evaluation and results are reported on in Section IV. We conclude in Section V.

*While visiting.

978-1-5090-3216-7/16/ $31.00

II. Background

ICEMAN uses the European Union project Haggle [7] as an integration framework, into which it adds implementations of multiple content dissemination algorithms, proactive and reactive utility-based caching, context-aware network coding, multi-authority node certification, access control, and interest privacy protection.

Participants: ICEMAN has three primary roles for nodes in the network: publishers, subscribers, and brokers. Publisher nodes add content with descriptive tags to the network. Subscriber nodes periodically broadcast node descriptions that include their interests. These descriptions are used by other nodes to identify which content matches a remote node's interests. Broker nodes forward content between publishers and subscribers, based on matches between the content tags and node interests.

Data Objects: Participants in ICEMAN share and receive data in units called *data objects*. Each such object has metadata associated with it, including a set of *tags* that are key-value attributes used to describe the content. Tags in the metadata are used to make forwarding decisions in the network. Content (such as a file) is inserted in a data object as its payload. A data object contains other metadata as well, such as the timestamp of when it was created, and a globally unique identifier, derived by computing a hash of its content and tags.

Data objects can also be exchanged between nodes without any payload. These data objects are announcements from a node to the network – for example, a node can send a *node description* that defines its interests to its neighbors. Nodes periodically retransmit their node descriptions to refresh their neighbors' view of their interests. Each node maintains a knowledge base of other nodes' cached content and interests.

Trust Model: ICEMAN supports the use of multiple concurrent authorities. This enables it to be resilient to failures of individual authorities and partitions of the network. Any node can serve as an authority, as long as other nodes agree to trust it. A node accepts another as an authority by receiving a shared secret from it out-of-band. This serves as the basis for securing messages from nodes to and from the authority. A node can send a Security Data Request ($SDReq$) to an authority to utilize its certification or authorization services. The authority uses a Security Data Response ($SDRes$) to return credentials to a node. These requests and responses are called Security Data Objects (SDOs) and are routed in the same manner as other data objects in the network.

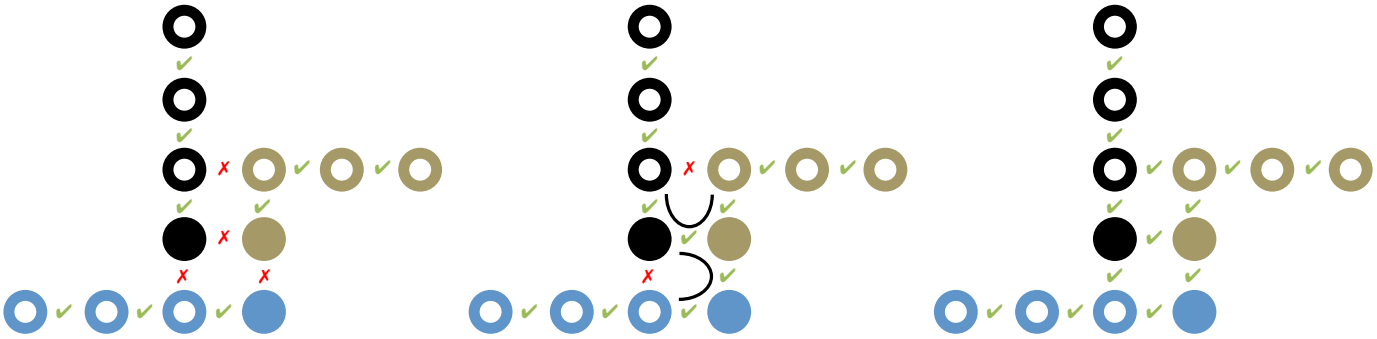
Authenticating Nodes: A node must choose to trust at least one authority in order to start participating in the network. As previously mentioned, the initial trust relationship between a node and an authority is established out-of-band. The resulting shared secret is then used to securely ask the authority to certify the node's identity certificate. The availability of multiple authorities increases the robustness of the system since each node can be certified by any authority.

When a node joins the network, it sends a self-signed identity certificate (in an $SDReq$) to the authority for signing. If the authority is configured to trust that node, it sends a signed version back (in an $SDRes$). Nodes only accept content from their neighbors if they are co-certified – that is, they share at least one trusted authority. Nodes $Alice$ and Bob exchange their certificates when they communicate with each other for the first time. If $C_{Alice,\alpha}$ is the set of certificates that $Alice$ has been issued by the set of authorities α, and $C_{Bob,\beta}$ is the set of certificates that Bob has been issued by set of authorities β, then trust is established if $\alpha \cap \beta \neq \emptyset$. This prevents an untrusted node from injecting content into the network.

Bootstrapping Trust: Nodes can join the network at any point in time and dynamically request that their identity certificates be signed by trusted authorities. If a node $Alice$ tries to join the network and there is only one node Bob in its vicinity, Bob cannot assume that $Alice$ has already interacted with a trusted authority to obtain a signed identity certificate. In particular, it is possible that Bob is the only node that $Alice$ can communicate with at the outset. Consequently, if Bob is not one of $Alice$'s trusted authorities, then Bob must relay communication from $Alice$ to authorities; otherwise, $Alice$ would not be able to get certified. ICEMAN addresses this trust bootstrapping conundrum by granting each node, such as $Alice$, a temporary window during which its $SDReq$ objects will be forwarded even though the node has not yet been co-certified. If $Alice$ receives a response from an authority in this window, it can join the network by including its signed certificate in future interactions. If the window expires without $Alice$ receiving a response, future data objects from it will be ignored by the network.

Content Routing: Data routing decisions are made by comparing a piece of content's tags with the interests of other nodes. Subscribers propagate their node descriptions containing a list of interests to other nodes that they encounter. Similarly, publishers share content with interested nodes that they encounter. Nodes aggregating the interests of others can serve as brokers to facilitate the hop-by-hop transportation of data objects from publishers to subscribers. Content caching in ICEMAN can either be *proactive* or *reactive* [8]. In proactive mode, a node pushes data objects to its neighbors based on the expectation that they will be interested in them in the future. In reactive forwarding, on the other hand, a node only sends a data object if the other node's *interests* and the content *tags* exceed a *matching threshold*.

Among other routing schemes, ICEMAN supports the use of a modified version of the DIsruption REsilient Content Transport (DIRECT) interest-driven content dissemination algorithm [13]. Every node periodically informs its neighbors about its interests by sending a timestamped node description. This is then propagated further in the network so that more nodes can respond if they have matching content cached. When a match occurs, the data object is forwarded along the reverse path – that is, to the neighbor from which the interest was first received. It is worth noting that interests are explicitly listed in a node description, which is then propagated through the network. This introduces the potential for significant dissemination of privacy-sensitive information.



(a) Scenario 1: Intra-group trust. (b) Scenario 2: Inter-group trust. (c) Scenario 3: Security not enabled.

Fig. 1: Three different co-certification scenarios are depicted. Authority nodes are represented with solid circles, while ordinary nodes are represented with rings. Only adjacent nodes are physically close enough to communicate directly.

Attribute-based Encryption: ICEMAN uses discretionary access control to define which nodes can read published content. This is enforced by limiting access to the payload of a data object using the *multi-authority* variant (MA-ABE) [6] of *attribute-based encryption* (ABE) [10]. Since MA-ABE is a type of *ciphertext-policy attribute-based encryption* [3], it embeds the access policy directly into the ciphertext. This is particularly useful in settings where potential subscribers are not known at the time of publication. Every publisher can encrypt each piece of content with a different access control policy. The ability of a node to decrypt a piece of content depends on the set of cryptographic attributes it has received from the authorities. Authorities are expected to issue credentials that correspond to the real-world characteristics of a node (such as its owner's organization, position, or role).

Content Access Control: Publishers encrypt content with an access policy before sending it to a remote node. The policy specifies which nodes can access the data, or more specifically, which combination of attributes are required to gain access. Note that each authority has a unique identifier, which determines the set of attributes for which it can issue encryption and decryption keys. The name of each attribute links it to the authority that issued it. With this approach, each publisher can construct a policy for its data requiring that any party that can decrypt the ciphertext has to possess a set of attributes issued by authorities that the publisher trusts. The publisher needs to know only the attributes that it uses in its policy. Each node requests encryption and decryption attributes from the authorities that it has established trust relationships with. It uses the shared secret key with the authority to establish secure communication for these requests. Nodes may request encryption and decryption attributes either on demand as they are needed to encrypt and decrypt content or through pre-provisioning – that is, requesting encryption and decryption attributes as soon as they join the network.

Metadata Access Control: Similar approaches are used by publishers and subscribers to limit the set of nodes that can act as brokers for their content and interests, respectively. A publisher encrypts each content tag with a policy that specifies which nodes are allowed to serve as brokers. Similarly, subscribers encrypt each interest with a policy that specifies which nodes can serve as brokers on their behalf. It is worth noting the flexibility of this framework, which allows each content tag and subscriber interest to be protected with an independent access policy. When a node receives a data object, it attempts to decrypt as many content tags as it can. Similarly, when it receives a node description, it attempts to decrypt as many interests as it can. Using the decrypted tags and interests, the node attempts to check if there is a sufficient match to forward the data object toward the subscriber.

III. Case Studies

The content dissemination and protection strategies in an ICN can interact in complex ways. We conducted a series of experiments to measure the effects of authorization, privacy, and caching policies on the efficiency and usability of ICEMAN. Below we report on three tradeoffs that we identified empirically:

Impact of Authentication on Routing: If all the nodes in a network act fairly and all the edges have the same bandwidth, the shortest path between any two nodes would be the optimal path for communication. However, as previously mentioned, a node trusts another node only if there is at least one authority that has certified both of them. If two nodes do not have such a trust relationship, the direct path between them cannot be used for any content exchange. This can lead to an increase in latency as the next optimal path may require more hops. Furthermore, this phenomenon can also partition the network graph into disconnected components, where no communication is possible between subsets of nodes. Greater trust increases routing efficiency at the cost of leaving the system increasingly vulnerable.

Impact of Caching on Authorization: ICEMAN uses encryption to ensure the confidentiality of data and to limit the nodes that can serve as brokers for content and interests. The efficiency with which a node can check for matches between a content's tags and a subscriber's interests depends on the extent to which the node can decrypt them. This may require the node to obtain cryptographic attributes from authorities, using SDOs. However, this process is complicated

by the fact that these objects are subject to caching policies at intermediate nodes between a requester and an authority. As storage pressure increases at a node, the SDOs may be evicted, adversely affecting authorization efficiency of remote nodes.

Impact of Privacy on Delivery Rate: An ICN router has access to content tags and interests in plaintext. ICEMAN addresses the privacy concerns of publishers and subscribers by allowing them to scope which nodes have access to the tags and interests, respectively. As more restrictive access policies are utilized, the privacy of publishers and subscribers increases. However, this also limits the nodes that can serve as brokers, bringing a concomitant reduction in routing robustness and efficiency.

IV. Evaluation

To ensure the repeatability of our experimental evaluation, we used the U.S. Naval Research Laboratory's Common Open Research Emulator (CORE 4.3) and Extendable Mobile Ad-hoc Network Emulator (EMANE 0.7.3) frameworks. Each ICN node's entire user-space code is run unmodified in a separate Linux (lightweight virtualization) *container* provided by CORE. The creation and movement of ICN nodes, publication of content, and subscription to interests, are orchestrated with scripts on each node, using CORE and EMANE programming interfaces.

A. Authentication / Routing Tradeoff

1) Goal – Understanding the baseline: Our first set of experiments was conducted to measure the effect of different configurations of trust among nodes on the time it takes to successfully deliver published data objects to interested subscribers. Figure 1 shows the physical arrangement of nodes in the experiment. Nodes are arranged in three groups, where each group has four nodes with one of the nodes an authority node (depicted with a solid circle). Nodes within a group trust each other as their identity certificates have been attested by the same authority.

Experimental Setup: We considered three scenarios:

1) *Intra-group trust* is established (as shown in Figure 1a). This case deals with the situation where there can be no data object exchanged across groups as no two nodes from different groups trust each other (as they are not co-certified).
2) *Inter-group trust* is established by letting all authority nodes in all groups co-certify each other. As shown in Figure 1b, this allows communication between groups but a data object may not follow the shortest path to a destination node.
3) *Ubiquitous trust* is shown in Figure 1c, where security is not enabled. Nodes are no longer required to be co-certified to exchange data objects. Hence, exchanges happen along the most optimal path permitted by the routing protocol.

Application Workload: In this experiment, 15 data objects were published and 31 subscriptions were issued. These publications and subscriptions were distributed among all groups and a total of 348 data objects could be delivered in the network (with multiple published data objects containing the same tags). All data objects were published and interests subscribed to within first 30 seconds of nodes initializing. Each data object had a payload that 512 KB in size.

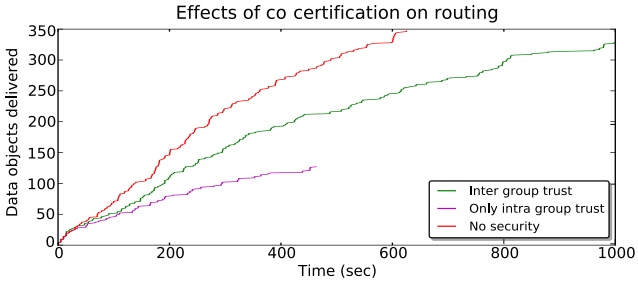


Fig. 2: Data objects delivered as a function of time. Individual plots correspond to different co-certification configurations.

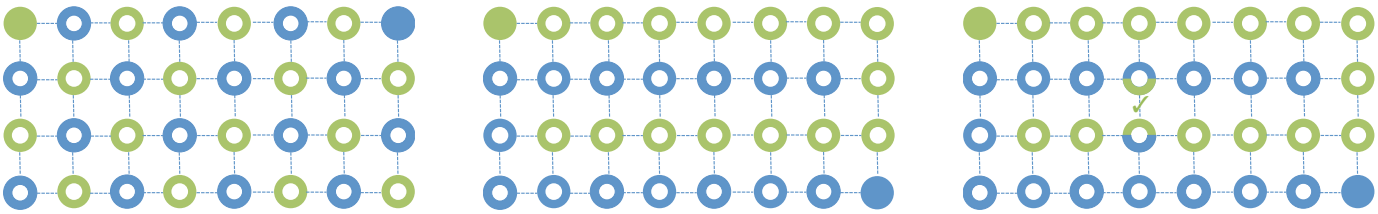
Results: Figure 2 shows the effect of various security policies on the performance of the data delivery. Lack of inter-group trust severely hampers data object dissemination across groups. This causes total network delivery to reduce significantly. An intermediate configuration allows inter-group communication but forces data objects to follow longer paths. This causes slower data delivery. Furthermore, a small fraction of data objects remain undelivered at the end of the experiment. Finally, a ubiquitous trust environment (where security is entirely disabled) removes any barriers for data object flow between nodes. This case is characterized by complete and fastest data object delivery across the network.

We can infer from this that if an ICN is designed for a public domain (such as use by first responders) where the primary focus is complete access to all data objects, the ICN should allow communication without the restriction of co-certification. On the other hand, enterprise and private ICNs, where data confidentiality is a higher priority, should either have a single authority or redundant authorities that certify all the nodes. For an option that allows a better balance between utility and security, please see the next section.

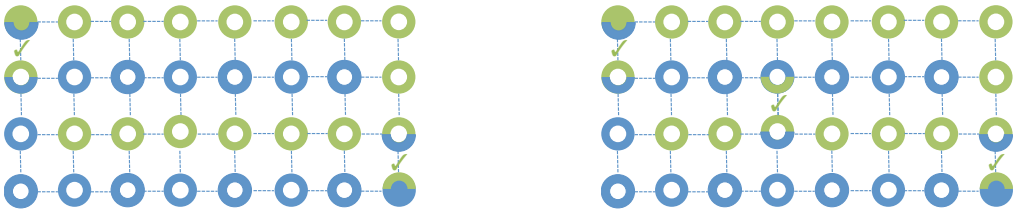
2) Goal – Understanding the benefit of bridge nodes: A node that has been co-certified by two or more authorities is trusted by the corresponding co-certification groups. It is therefore able to bridge the groups by forwarding data objects from one group to the others. We empirically study the effect of the presence of such bridge nodes on content delivery performance.

Experimental Setup: We considered the following five scenarios:

1) *Untrusted neighbors* are depicted in Figure 3a. In this scenario, each node is a member of one of two distinct trust groups. Each group has been separately co-certified



(a) Each node's neighbors are untrusted. (b) Longest linear distinct trust groups. (c) Single bridge between two trust groups.



(d) Two bridges between two trust groups. (e) Three bridges between two groups.

Fig. 3: Co-certification creates trust groups. Five different configurations are shown here. Solid circles are authorities. Rings are ordinary nodes. Only adjacent nodes are close enough to communicate.

by a different authority. The nodes are arranged so that no two neighbors are from the same group. Hence, every node is surrounded by untrusted nodes.

2) *Longest linear groups* are shown in Figure 3b. Nodes in this setting are arranged to maximize the number of hops a data object can travel while remaining within a single trust group. The configuration is referred to as *maximum piping*.
3) *Single bridge* is a refinement of the previous case. The difference is the presence of a single bridge node, as depicted in Figure 3c.
4) *Two bridges* are illustrated in Figure 3d. The second bridge reduces (on average) the number of hops a data object must traverse to be able to cross from one trust group to another.
5) *Three bridges* are shown in Figure 3e. As expected, the availability of a third bridge further reduces the intra-group number of hops a data object must traverse before crossing into another trust group.

Application Workload: To facilitate a consistent comparison with the baseline, the same data object publication and interest subscription patterns were used as those described in Section IV-A1.

Results: Figure 4 reports the data delivery achieved (as a function of time) when different node trust configurations are utilized. As expected, disabling security completely results in the best data delivery (in the absence of any adversaries). To limit the attack surface of the ICN, the use of co-certification is recommended (to ensure outsiders cannot inject content into the system). In the baseline case we saw this impose a high negative impact on performance.

Our results here show that we can achieve both high performance and low risk through the judicious use of sufficient

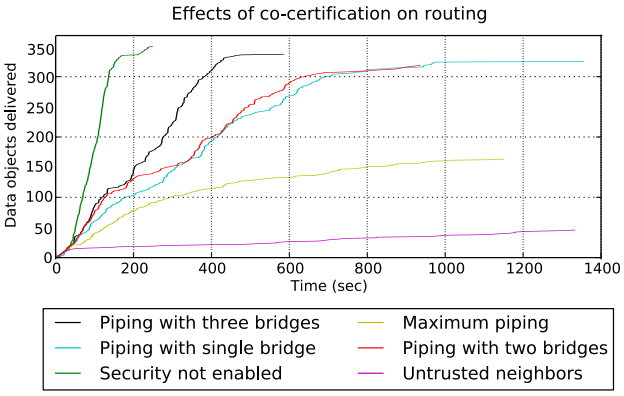


Fig. 4: Data objects delivered as a function of time. Individual plots correspond to different co-certification configurations.

bridge nodes. In particular, when three bridge nodes were used, we saw data delivery performance approaching the case when security was disabled. As the number of bridge nodes decreases to two and then to one, we continue to observe all the data objects delivered but over longer timespans.

In the case of maximum piping, data objects can only travel within a single trust group. Consequently, the maximum number of deliveries is limited by what can be retrieved without objects crossing over from the other group. Similarly, in the case that all neighbors are untrusted, the only objects delivered are from a node to itself. This occurs if one application on the node publishes a piece of content while the interests of another application match the content's tags.

Our findings provide a prescription for designers of ICNs with multiple trust domains. In this setting, it is critical to introduce enough bridge nodes in the architecture. While it

is well understood that increasing the number of such nodes is important for reliability, our results show the direct and significant effect on improving data delivery rates as well.

B. Authorization / Caching Tradeoff

1) Goal – Understanding the effect of caching policies on distributed access control: In the absence of dedicated control channels for security metadata, an ICN must utilize the underlying data forwarding infrastructure to send security-related requests and responses – that is, the SDOs described in Section II. Since access control in a distributed system is implemented through the use of SDOs from ordinary nodes to authorities and back, the caching policy at intermediate brokers determines the speed with which authorization requests complete.

Experimental Setup: We studied this question in a setting with the hourglass topology depicted in Figure 5. It contains two authorities α_1 and α_2 at opposite corners of the network. They are responsible for issuing encryption and decryption attributes to nodes that request them (and are authorized according to the authorities' configurations). The node β is the narrow waist of the hourglass. It separates the two sides of the network where α_1 and α_2 are located, respectively.

The bottleneck at node β facilitates controlled analysis of authorization performance as a function of the caching policy (at β). We consider the canonical case where a data object is published by node n_i with an access policy that contains an attribute from authority α_1. This data object has tags that match the interests of a node n_j on the opposite side of the network. Consequently, when n_j attempts to access the content, it will send a security data request that must traverse node β en route to the authority α_1. Similarly, the security data response from α_1 to n_j must go through β.

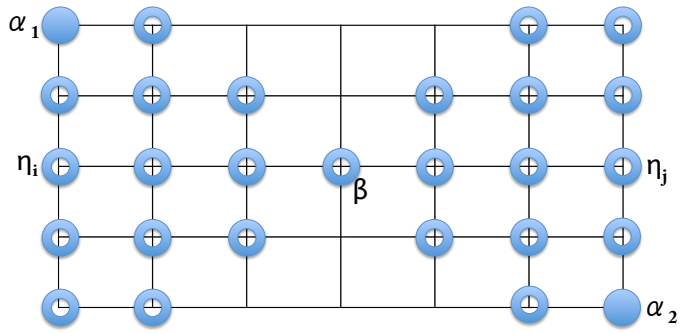


Fig. 5: Node β serves as a bridge between the groups of nodes on its left and right, credentialed by authorities α_1 and α_2, respectively.

We compared four representative caching strategies to understand their effect on authorization performance:

1) *Ubiquitous SDO prioritization.* All nodes, including bridges, use caching policies that give SDOs highest priority, minimizing the chance that they will be dropped at an intermediate broker en route to or from an authority.
2) *Only bridges are SDO-agnostic.* All nodes, except bridges, use caching policies that prioritize SDOs. This configuration models the case where bridges minimize per-object analysis to be able maximize throughput. In this setting, bridges will not distinguish between SDOs and other data objects.
3) *Only bridges prioritize SDOs.* In practice, the importance and low volume of SDOs may argue for bridges to be configured with caching policies that prioritize SDOs. This scenario assumes this but examines what happens when the remaining nodes do not perform similar prioritization.
4) *No SDO prioritization.* The final case provides a baseline for comparison. This scenario represents the ICN performance if it does not build in any cognizance of the effect of caching on authorization.

Application Workload: To saturate the cache at the bridge β, 8 data objects were published at the node. Each of these had content tags that matched interest subscriptions from the bridge. The 13 nodes on each side of the bridge each publish a single data object, for a total of 2 x 13. Each is encrypted with an access policy that contains a unique attribute that can be obtained from the authority that is reachable without crossing the bridge. Each node expresses an interest that matches an object that must traverse the bridge to arrive at the subscriber. Further, subscriptions are distributed across the network to balance the delivery overhead at all nodes.

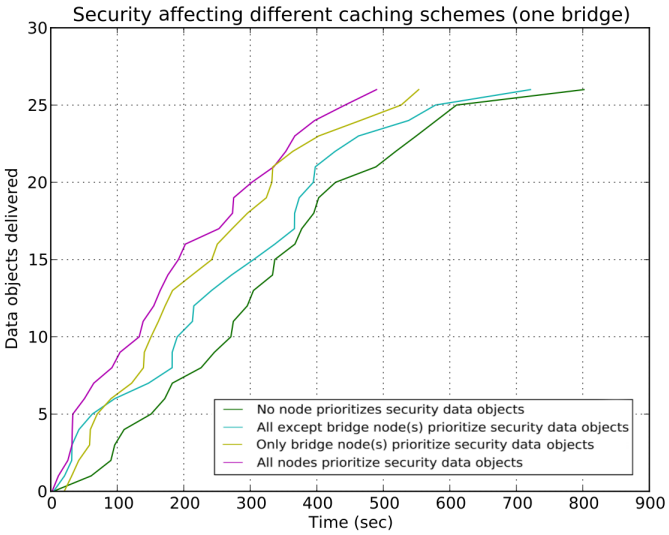


Fig. 6: Access control performance depends on the prioritization of SDOs in caching policies at nodes.

Results: Figure 6 describes how many objects have been delivered as a function of time. The four plots correspond to the different caching policy configurations described above. Recall that every data object's delivery is predicated on the completion of an authorization operation (that retrieves a cryptographic attribute from an authority on the opposite side of the bridge). So these plots are reporting on the effect of the caching policies on access control performance.

When all the nodes in the system prioritize SDOs, authorization and consequently data object delivery, completes fastest. When none of the nodes do so, performance is the worst. Interestingly, the prioritization of SDOs at the single bridge node does more to improve performance than such implementing this caching policy at all the remaining nodes in the system. This derives from the fact that the bridge is the bottleneck in communication.

In addition to affecting the average performance, caching policies also impact the variance in the performance. Configurations with policies that result in worse performance also result in higher standard deviations, as seen by the error bars in Figure 7.

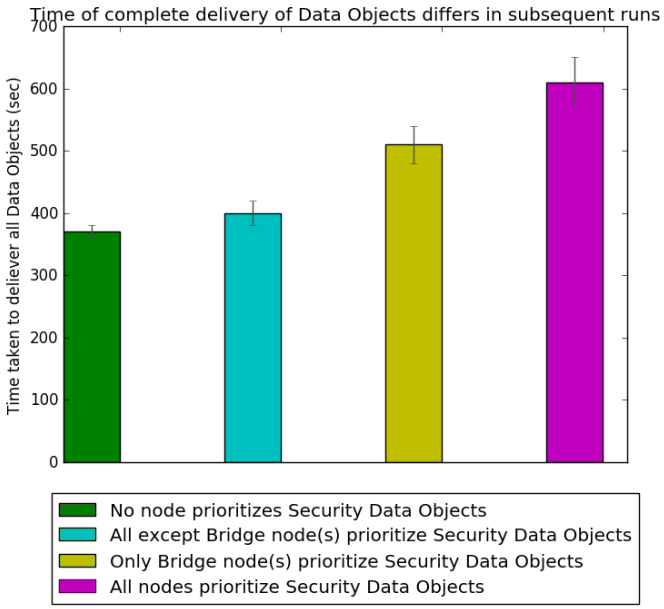


Fig. 7: Caching policies that result in longer times for access control requests to complete have the side-effect of larger variations in performance as well.

C. Privacy / Delivery Tradeoff

1) Goal – Pricing privacy primitives: ICEMAN protects the privacy of publishers and subscribers by allowing them to encrypt their content tags and interests, respectively. This encryption is effected using MA-ABE policies that scope the set of nodes that will have access to the tags and interests. Coarser policies result in lower privacy. Finer-grained policies protect the privacy of publishers / subscribers. To understand the price being paid for privacy, we measured the associated cryptographic costs.

Experimental Setup: We constructed the following micro-benchmarks:

1) *Encryption* measures the time for symmetric encryption of a tag or interest.
2) *Decryption* measures the time for symmetric decryption of a tag or interest.
3) *Capability generation* measures the time to encrypt a symmetric key with an MA-ABE access policy.
4) *Capability usage* measures the time to decrypt a symmetric key encrypted under an MA-ABE policy.

Workload: The set of micro-benchmarks was run, while varying the number of attributes used in the MA-ABE access policy. More attributes correspond to a more specific policy, which results in reduced privacy leakage.

Results: Figure 8 reports the increasing costs of more specific access policies. A runtime cost is introduced that results in each node taking longer to be able to access content tags and subscriber interests. The cost is commensurate with the number of attributes utilized.

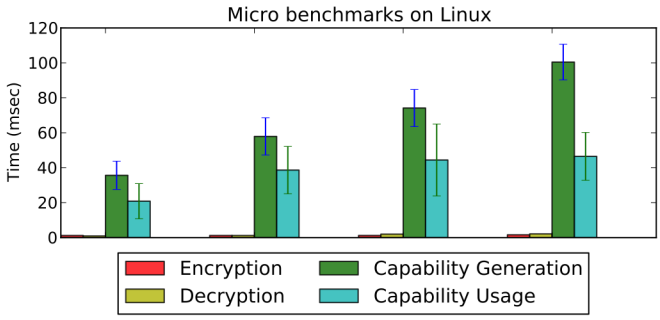


Fig. 8: As the number MA-ABE attributes used increases, the capability generation and usage times grow.

2) Goal – Pricing privacy-sensitive routing: A publisher can preserve their own privacy by limiting the nodes in the ICN that can see the tags they label their content with. Similarly, subscribers can maintain their privacy by scoping which nodes can see their interests. The consequence is that only some nodes can serve as brokers. To understand the impact these privacy protections have on the data delivery rates in the ICN, we conducted the following study.

Experimental Setup: The setting for this analysis was the 4 x 4 grid of nodes depicted in Figure 9. i, j, and k denote three groups of subscriber nodes. The nodes in i are also publishers. In this context, we studied the following three scenarios:

1) *Circumference routing* where the tags encrypted by nodes from group i can only be decrypted by nodes from groups i, j, and k.
2) *Universal routing* where all nodes have sufficient decryption attributes to be able to match and forward data objects.
3) *Promiscuous routing* where privacy protections are disabled.

Application Workload: Within the first 20 seconds after node initialization, they publish content or subscribe to specific interests. Each publisher shares 4 data objects, while subscribers express interests that match the tags on these objects. A maximum of 160 data objects can be delivered in this setting.

Results: If security is disabled, promiscuous routing results. In particular, both the tags of all content and the interests

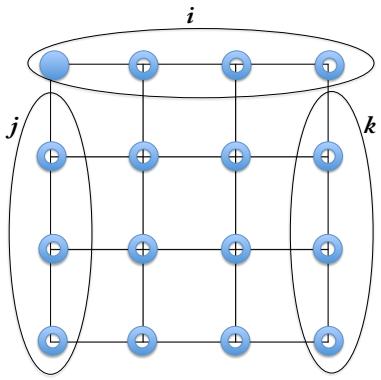


Fig. 9: Three levels of routing privacy protection were studied using this arrangement of nodes. They differed in which nodes could serve as brokers and the cost for doing so.

of subscribers are exposed for all intermediate nodes to see. This mode of operation provides no privacy protection but does result in the fastest delivery of content. Note that all data 160 objects are delivered by the 50 second mark, as seen in Figure 10.

In the case of universal routing, publishers apply protection to content tags, subscribers do the same for interests, and all intermediate nodes are provided with attributes that allow them to access content tags and interests. While privacy protections are used, all nodes are trusted enough to be able to serve as brokers. This allows multiple paths for content to flow from each publisher to each subscriber. The result is that data objects are all delivered by the 70 second mark. The reduction in speed is the result of brokers needing to decrypt tags and interests to perform matches.

Finally, circumference routing occurs when nodes outside groups i, j, and k are not trusted to access the private tags and interests of publishers and subscribers, respectively. The untrusted nodes cannot serve as brokers, preventing content from flowing through them. Consequently, content must be "piped" through a longer path. This results in the data objects taking close to 100 seconds to be delivered. The cost of maintaining tag and interest privacy is the slower delivery time.

V. Conclusion

Using the SRI's open source ICEMAN implementation of an ICN that operates over a MANET, we examined three trade-offs between maintaining the privacy of content publishers and subscribers on the one hand, and the performance of routing and content delivery on the other. In particular, we empirically demonstrated that (i) adding authentication to prevent outsiders from injecting content in the ICN comes at a cost for routing performance, (ii) using the data plane of an ICN for security-related communication requires corresponding cache policy prioritization, and (iii) the use of more privacy-sensitive content tags and subscriber interests increases the delivery time of affected content. We also make specific prescriptions for ICN architectures, including the use of sufficient bridge nodes, specific cache policy priorities, and relaxing privacy protection of metadata when possible.

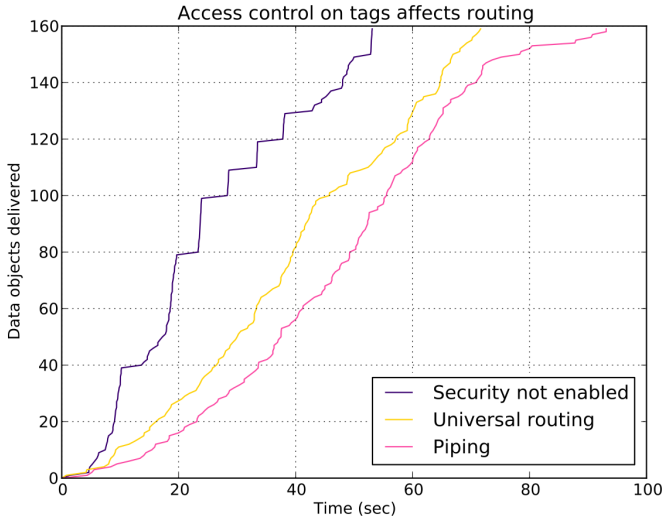


Fig. 10: Access control on tags and attributes slows down content delivery.

References

[1] Alexander Afanasyev, Priya Mahadevan, Ilya Moiseenko, Ersin Uzun, and Lixia Zhang, Interest flooding attack and countermeasures in Named Data Networking, *12th IFIP Networking Conference*, 2013.

[2] Bengt Ahlgren, Christian Dannewitz, Claudio Imbrenda, Dirk Kutscher, and Borje Ohlman, A survey of information-centric networking, *IEEE Communications Magazine*, Vol. 50(7), 2012.

[3] John Bethencourt, Amit Sahai, and Brent Waters, Ciphertext-policy attribute-based encryption, *28th IEEE Symposium on Security and Privacy*, 2006.

[4] Mihaela Ion, Jianqing Zhang, and Eve Schooler, Toward content-centric privacy in ICN: Attribute-based encryption and routing, *3rd ACM SIGCOMM Workshop on Information-Centric Networking*, 2013.

[5] Jun Kuriharay, Ersin Uzun, and Christopher Wood, An encryption-based access control framework for content-centric networking, *14th IFIP Networking Conference*, 2015.

[6] Allison Lewko and Brent Waters, Decentralizing attribute-based encryption, *30th International Conference on the Theory and Applications of Cryptographic Techniques*, Springer, 2011.

[7] Erik Nordstrom, Christian Rohner, and Per Gunningberg, Haggle: Opportunistic mobile content sharing using search, *Computer Communications*, Vol. 48, Elsevier, 2014.

[8] Soon Oh, Davide Lau, and Mario Gerla, CCN in tactical and emergency MANETs, *3rd IFIP Wireless Days Conference*, 2010.

[9] Mariana Raykova, Hasnain Lakhani, Hasanat Kazmi, and Ashish Gehani, Decentralized authorization and privacy-enhanced routing for ICNs, *31st Annual Computer Security Applications Conference*, 2015.

[10] Amit Sahai and Brent Waters, Fuzzy identity-based encryption, *24th International Conference on the Theory and Applications of Cryptographic Techniques*, Springer, 2005.

[11] Abdullatif Shikfa, Melek Onen, and Refik Molva, Bootstrapping security associations in opportunistic networks, *6th International Workshop on Mobile Peer-to-Peer Computing*, 2010.

[12] Abdullatif Shikfa, Melek Onen, and Refik Molva, Privacy and confidentiality in context-based and epidemic forwarding, *Computer Communications*, Elsevier, Vol. 33(13), 2010.

[13] Ignacio Solis and J. J. Garcia-Luna-Aceves, Robust content dissemination in disrupted environments, *3rd ACM Workshop on Challenged Networks*, 2008.

[14] Reza Tourani, Travis Mick, Satyajayant Misra, and Gaurav Panwar, Security, privacy, and access control in Information-Centric Networking: A survey, *arXiv:1603.03409*, 2016.

[15] Samuel Wood, James Mathewson, Joshua Joy, Mark-Oliver Stehr, Minyoung Kim, Ashish Gehani, Mario Gerla, Hamid Sadjadpour, and J.J. Garcia-Luna-Aceves, ICEMAN: A system for efficient, robust and secure situational awareness at the network edge, *32nd IEEE Military Communications Conference*, 2013.