Routing and Efficiency in Mobile Ad-hoc networks:Trade-offs along the security continuum

***Abstract*—ICEMAN is an Information Centric Network for Mobile Ad hoc Network. It provides a substrate for applications to publish content and subscribe to their interests in challenged networking conditions, where disruptions in connectivity to peers, changes in network topology, and resource limitations are the norm. Since the content and interests are brokered by peers in the system, ICEMAN allows (i) authorities to define mutually trusted groups, (ii) publishers to scope which nodes can access their content, and (iii) subscribers to specify brokers that can act on their behalf. These three mechanisms enhance security at the cost of limiting the space of peer-to-peer routes that content can flow through. In this paper, we explore the tradeoff between efficiency and the security and privacy of data and objects in ICEMAN. Our findings have important implications for efforts to secure ICNs and we show that there is an inherent interplay between the need to have more specific information for efficient routing decisions and the need to ensure trust and confidentiality within such a decentralized system.**

I. Introduction

Information-centric networking is a new approach for content distribution and retrieval that has drawn considerable attention in recent years. While there are several competing architectures and implementations, the general idea is that data is de-coupled from location and network functions are driven by named-content instead of the traditional source-destination based models. At a high-level, publishers can advertise their content and their descriptions, subscribers can advertise their interests and objects can be accessed by specifying their names. Data transport or routing decisions are content-aware, driven by matches between interests of nodes and the descriptions of the published content. The network can also take advantage of various performance optimizations such as in-network caching of content etc in order to reduce latency and improve efficiency. Despite several desirable properties, important concerns related to trust , privacy and security of the participants need to be addressed.

ICN based architectures are typically geared towards mobile environments where nodes can join or leave the network at will. The ability to authenticate ICN nodes in the absence of a single central trusted authority becomes really important for the secure operation of the network. In absence of authentication, a malicious node can not only generate and consume content but it can also attempt to overwhelm the network using resource exhaustion type attacks. Data and privacy breaches are another important concern where queries and forwarding mechanisms can reveal sensitive information about publishers, subscribers and content meta-data [**?**]. Descriptive information is usually embedded in the meta data associated with a data object. ICN routing algorithms typically leverage these content descriptions and consumer interests for forwarding decisions. Current routing mechanisms leak sensitive information related to meta data and other aspects of data objects and nodes involved in the publishing and consumption of data objects.

Several approaches have been suggested in the literature to improve the security, privacy and confidentiality of ICN based publish-subscribe systems. Enhanced security and privacy leads to a much degraded network performance. The biggest impact on network performance is through reduced forwarding options at routers causing data objects to follow longer paths that eventually leads to lower data rates and degraded network performance. Reduction of forwarding options at routers can also lead to routing *black holes* for data objects: a data object might never be able to find a path to the subscriber. If data object delivery, the successful generation, transmission and consumption of data object, is adversely affected by poor security configuration, this can further degrade performance for network nodes. We present detailed analysis and results from a first of its kind measurement and modelling based study of security in ICEMAN, SRI International's implementation of ICN [13]. Specifically, we address:

The affect of trust on content routing A secure ICN architecture requires nodes to prove their identity before they can exchange data with other nodes. This is usually achieved by introducing a PKI in the network where some nodes can vouch for other nodes by authenticating their identities and their relationship to their public keys. This requirement however can severely limit the routing ability of a network.

The effect of in-network caching Typical ICN architecture use payload encryption on data objects in order to ensure confidentiality and security. While there are several competing approaches for encryption, ICEMAN uses a distributed access control mechanism that enables publishers to encrypt data objects using complex access policies citeWood13. In order to achieve encryption, a node has to negotiate encryption credentials for specific policies from trusted authorities. In the absence of a central authority, ICEMAN uses a multi-authority attribute-based encryption (MA-ABE) scheme [8] . The use of in-network content caching complicates the issue. Caching is used to reduce latency and bandwidth however, the caching algorithm also needs to handle the communication between the nodes and the trusted authorities for credentials which can adversely effect network throughput.

Effect of encryption on routing An ICN router has access to content tags (from publishers) and content interests (from subscribers) in plain text. ICEMAN addresses privacy concerns of publishers and subscribers by providing network-wide ability to set access policy on tags and interests. Therefore, only those routers that are provisioned to decrypt these tags and interests can participate in forwarding the data. As noted earlier, this can cause data objects to follow longer paths that can leads to degraded performance.

In this paper, we explore the tradeoff between efficiency and the security and privacy of data and objects in information centric networks. Our findings have important implications for efforts to secure ICNs and we show that there is an inherent interplay between the need to have more specific information for efficient routing decisions and the need to ensure trust and confidentiality within such a decentralized system.

Paper Organization. The rest of the paper is organized as follows. Section II describes ICEMAN and its privacy-enhancing architecture in detail. We describe our simulation test-bed and experimental methodology in Section IV . We analyse our results and summarise the key findings in Section V and conclude in Section VI.

II. Background

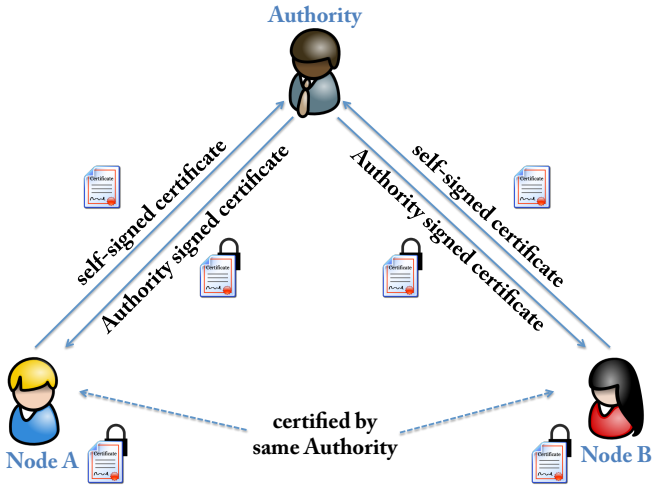


Fig. 1: Co-certification between two nodes in presence of one authority.

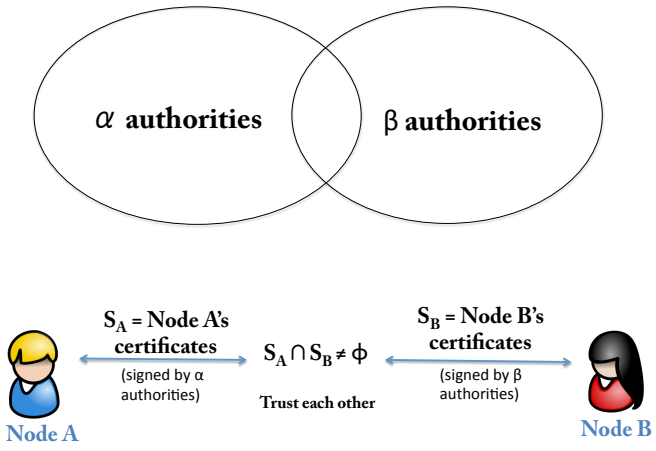


Fig. 2: Successfully co-certificating between two nodes depends on presence of mutually trusted authorities.

Our work is based on SRI International's Information CEntric Mobile Ad hoc Network (ICEMAN) implementation of ICN [13]. ICEMAN builds on top of Haggle [9] to provide in-network caching of data objects, distributed search in these caches, dynamic binding and temporal decoupling. It is an event-based and layer-less architecture where multiple independent managers running in separate threads communicate asynchronously through events. These managers further spawn asynchronous threads to perform computationally expensive operations. Each manager is tasked with providing a specific service, e.g. security manager is responsible for providing all security related services like encryption and decryption of content, content description (*tags*) and subscriber's *interests*. Moreover, security manager also signs identification certificates issued by nodes in order to authenticate their identities.

ICEMAN treats a data object, O as unit of abstraction. A data objects has meta data, $M(O)$ that essentially is a set of key-value attributes that are used to associate information with a data object. Content can be attached to a data object as payload $P(O)$ (which is usually in form of a file). Furthermore, a data object has a creation time stamp, $TS(O)$ and a globally unique identifier, $ID(O)$ that is derived from SHA1 of data object.

$$O \equiv \{M(O)\}[P(O)] \vee \{M(O)\} \qquad \text{(data plane)}$$

In order to enable content dissemination and consumption, a parallel protocol to establish network plan is followed. This involves exchanges of data objects without any payload. These data objects are announcements from a node to the network, e.g. an application node can send a node description showing application's interests or a device node can send cache summary in node description to its neighbours. Nodes re-transmit their node description in order to keep their neighbours updated about the state of the node. Each node maintains a knowledge-base of the capabilities of other nodes; even if they are not immediate neighbour nodes.

$$O \equiv \{\{M(O)\} \qquad \text{(network plane)}$$

The security architecture The security architecture in ICEMAN relies on a Public Key Infrastructure (PKI) where some nodes are pre-configured to act as authorities. This allows participants to verify the origin and integrity of the content they receive. Nodes send Security Data Request ($SDReq$) to an authority requesting services offered by that authority. Authority uses Security Data Response ($SDRes$) to distribute credentials to nodes. These Security data objects SDO are exchanged as part of a network plane.

$$SDO \equiv SDReq \vee SDRes$$

$$SDReq \equiv \{M(SDReq)\}\ ,\ SDRes \equiv \{M(SDRes)\}$$

A. Authenticating Publishers

Every node in the system chooses its trusted authorities before it joins the network. The initial trust relationship between the nodes and authorities can be established out of band, for the purpose of this paper, we assume a network administrator can configure each node with a different shared secret key for each of the trusted authorities that it wants to talk to. The nodes can later use this shared secret to request identity certificates from the trusted authorities. The use of multiple trusted authorities and authority redundancy makes trust bootstrapping robust and flexible.

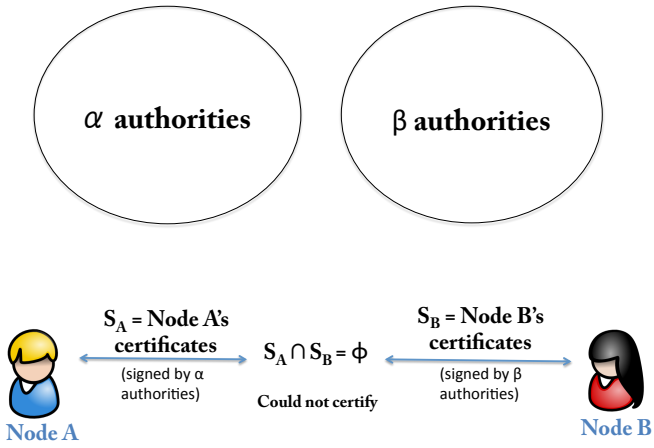


Fig. 3: Co-certification fails when two nodes can not show certificates signed by at least one mutual trusted authority.

When a node comes online, it sends a self-signed identity certificate, ($SDReq$) to the authority for signing. As the authority is configured to trust that node, it signs the identity certificate of the node and sends a signed version back ($SDRes$). Nodes only accept content from their neighboring nodes if they are co-certified ie. they share at least one certification authority. This prevents an unauthorized node from exchanging potentially harmful content with the network. Nodes $Alice$ and Bob exchange their certificates when they communicate with each other for the first time. If $C_{Alice,\alpha}$ are the certificates that $Alice$ has been issued by set of α authorities and $C_{Bob,\beta}$ are certificates that Bob has been issued by set of β authorities, then trust is established if:

$$\alpha \cap \beta \neq \emptyset$$

Similarly, if there are no overlapping authorities that have issued certificates to both $Alice$ and Bob, then Bob would not be co-certified by $Alice$

$$\alpha \cap \beta = \emptyset$$

Therefore a node trusts another node if and only if there is at least one authority that has certified both of them.

Bootstrapping trust:: Nodes can join the network anytime and dynamically request their certificates from trusted authorities. Lets assume 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 at this point. Its also possible that Bob is the only node that $Alice$ can communicate with for a substantial amount of time. If Bob is not an authority node, then Bob has to relay all communication from $Alice$ to an authority, or otherwise $Alice$ would not be able to request certificates from the authority. This raises another question, how can Bob start relaying data objects from $Alice$ to the network without being able to co-certify $Alice$. We use trust bootstrapping to answer this problem.

The basic premise behind trust bootstrapping is to temporarily accept self-signed certificates from a new node but limit communication to only Security Data objects (SDO). If a node $Alice$ fails to provide an authority signed identity certificate within a specified time-frame or *grace period*, further communication from that $Alice$ is ignored.

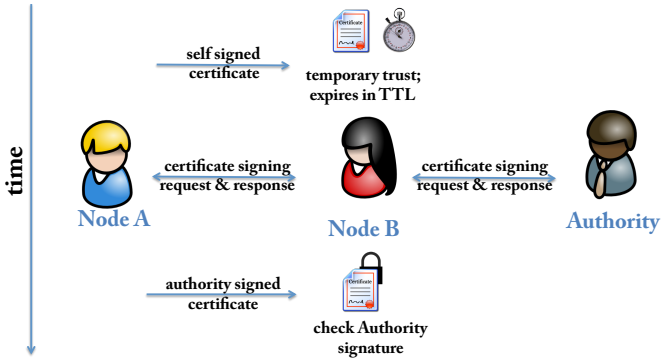


Fig. 4: Temporal trust that a relay node incurs in another node, enables communication between a node and a disjoint authority.

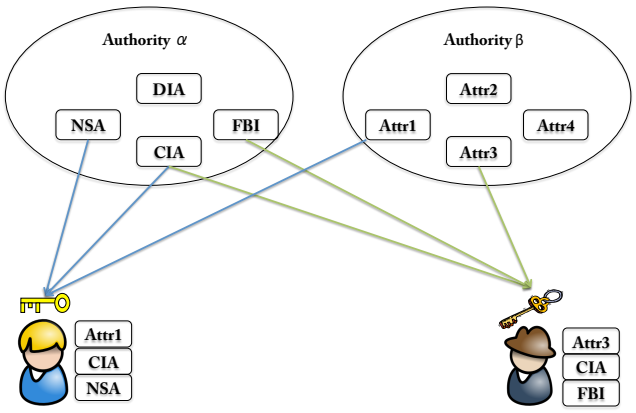
B. Authorizing Subscribers

ICN's decentralized architecture makes publishers unaware of potential subscribers of their content. In order to control the access of published content, we introduce discretionary access control on content. We achieved discretionary access control on content using multi-authority (MA-ABE) [8] version of attribute-based encryption (ABE).[11]

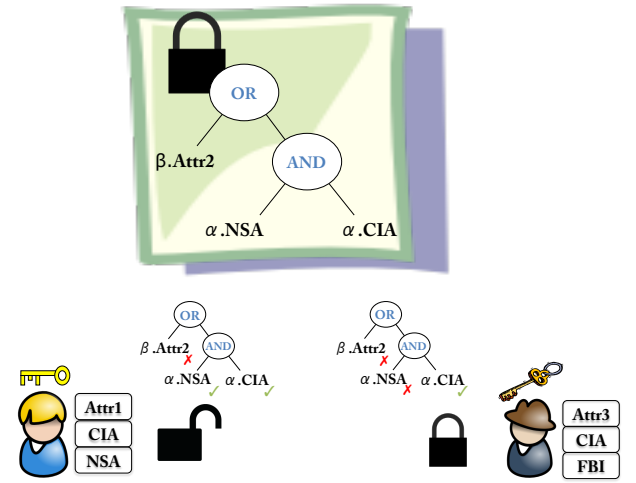
Attribute-based encryption (ABE)[11] can be used for protecting content in a setting where potential receivers are not known at the time of publication. The cipher-text policy attribute-based encryption [1] primitive embeds an access policy directly into each cipher-text, and associates a set of attributes with each decryption key. A key can decrypt a cipher-text if and only if its attributes satisfy the encryption policy. Nodes receive keys containing appropriate attributes (such as their organization, position, or role) from authorities. Thus, each publisher can encrypt each piece of content with a different access control policy that limits the set of nodes that can decrypt and learn the content based on their attributes.

Since we are dealing with a setting where it is difficult to agree on a single trusted authority that issues attributes for all users, we use multi-authority attribute-based encryption (MA-ABE) [8] that decentralizes the trust by having several independent authorities that can issue decryption keys corresponding to different attributes. This allows maximum flexibility for the publisher.

Publishers encrypt content with an access policy before sending it to a remote node. The policy specifies who can access the data, or more specifically what combination of attributes are required for gaining 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 cipher-text has to possess a set of attributes issued by authorities that the publisher



(a) Authorities distribute ABE encryption and decryption attributes to nodes. Here description attributes are being distributed.



(b) Publisher can encrypt payload using ABE-policy. Only those nodes that have decryption attributes can decrypt the payload.

Fig. 5: Multi-Authority Attribute Based Encryption in ICEMAN

trusts. The publisher needs to know only the attributes that it uses in its policy.

The distribution of the MA-ABE decryption keys is part of the ICN solution. Each node requests encryption and decryption attributes from the authorities that it has established trust relationship 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 need them to encrypt and decrypt content or with pre-provisioning – that is, requesting encryption and decryption attributes as soon as they join the network.

C. Selecting Brokers

A node can either be a publisher, subscriber, authority or broker in this architecture. Broker nodes facilitate the hop-by-hop transportation of data objects in ICEMAN based on content tags and node interests. Routing in ICEMAN can either be proactive or reactive [15]. In proactive mode, a node pushes all

```
.....
<metadata>
<interest name="content">
    Nuclear Launch Codes
</interest>
.....
</metadata>
.....
```

(a)

```
.....
<tag name="content">
    Nuclear Launch Codes
</tag>
.....
<data >..... </data>
```

(b)

Fig. 6: Parts of node description (a) and data object (b) XML. Interests and tags are included as raw text.

the data objects it receives to all of its neighbours. In reactive forwarding on the other hand, a node forwards data objects only to those nodes that match a certain criteria - e.g. *interests*. If the matched node is a neighbour, data objects are simply forwarded to that node. Otherwise, ICEMAN tries to determine a neighbour that can transport data object to remote node and forwards the data object to that neighbour. ICEMAN's uses a modified version of DARPA's DIsruption REsilient Content Transport (DIRECT) as an interest-driven content dissemination protocol. Every node can periodically inform other nodes about its interests by sending a timestamped node description with a list of interests embedded in it. Using either proactive or reactive strategies, this description is further replicated in the network so that more and more nodes can join the pool of nodes that can satisfy this data object request. Upon a data object match with an interest, the data object is forwarded along the reverse-path to the neighbor from which the interest was first received. It is important to observe that potentially all nodes in a network may have access to a node's description. As interests are listed as plain text, all participating nodes can 'see' it, making node description privacy insensitive.

As described earlier, data objects include descriptive tags that describe the content included in the objects. Even if the payload is encrypted, these published data objects can leak important information as they are in plain-text. A routing node: a node with access to both node description and a published data object needs to 'see' these in-order to make a routing decision. ICEMAN allows publishers and subscribers to use the same access policy used for content to restrict the nodes that can take routing decisions on their behalf as well. This helps prevent the leakage of privacy information through interests

and object-meta data etc.

For a publisher, access control on forwarding nodes is achieved by applying ABE encryption on content tags using the specified policy whereas a subscriber achieves similar capabilities by encrypting the interests in its node description. When a node replicates a data object to its neighbours, it first encrypts its tags or interests. A publisher can specify multiple tags with a data object it publishes. Similarly, a subscriber can set multiple interests in a node description it sends to the network. A publisher or subscriber can set different policies for each tag or interest. A node after receiving the data object tries to decrypt its tags or interests. In the process, it requests decryption attributes from the authority. If this node does not have the rights for the decryption attributes, it fails to determine if it needs this data object itself. If the node is configured to proactively disseminate, it will forward this data object to its neighbouring nodes, but in the case of reactive dissemination, it will not be able compare the data object across the database of node descriptions, resulting in halting the spread of data object from that node. It is important to node that unlike ABE encryption of data object's payload, every node that participates in relaying the data object must be privileged to decrypt the tags and interests of data object request and responses. A publisher can specify a different policy for content and forwarding nodes. Usually a publisher would set lenient policy for forwarding as compared to more stringent policy on content.

Another important aspect of data object dissemination is to distinguish between data object request and response. Some relying nodes will only be able to forward node descriptions (creating a node description scope) while some other nodes will only be able to forward data objects (creating data object scope). Furthermore, a handful of nodes will be able to perform both. Hence these will be nodes that can compare and forward data objects to the requesting node.

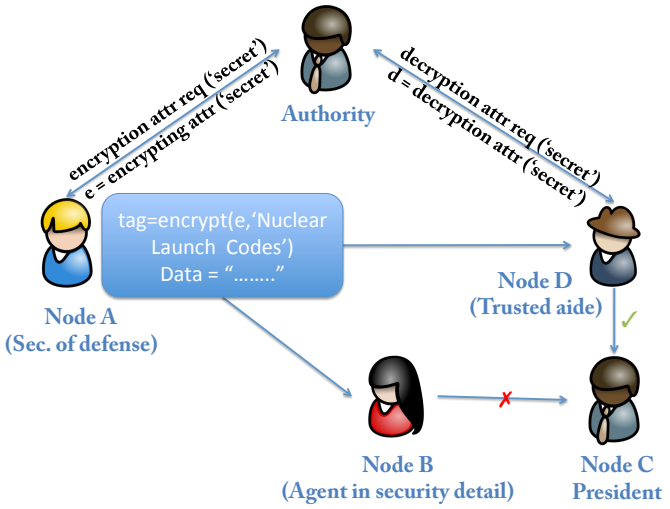


Fig. 7: A network administrator usually creates different access control levels among nodes and provisions nodes to receive encryption and decryption attributes from authorities.

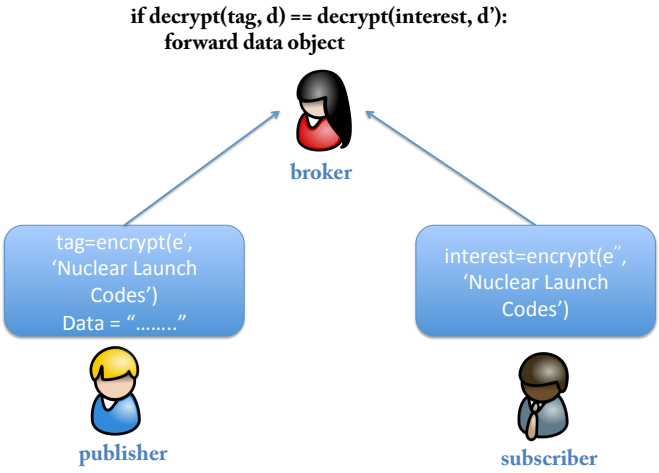


Fig. 8: A broker node has access to decryption attributes for both an interest and a tag. These decrypted tag and interest should also satisfy the matching criteria.

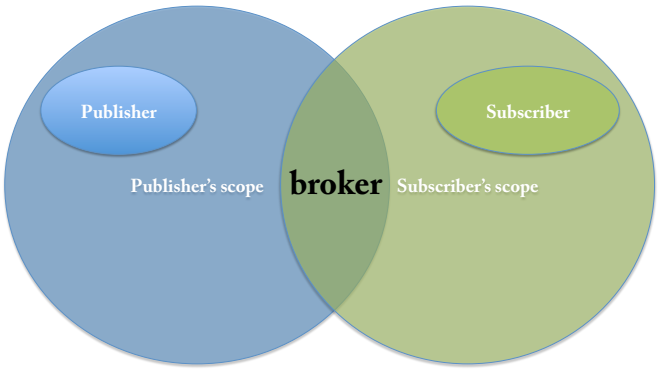


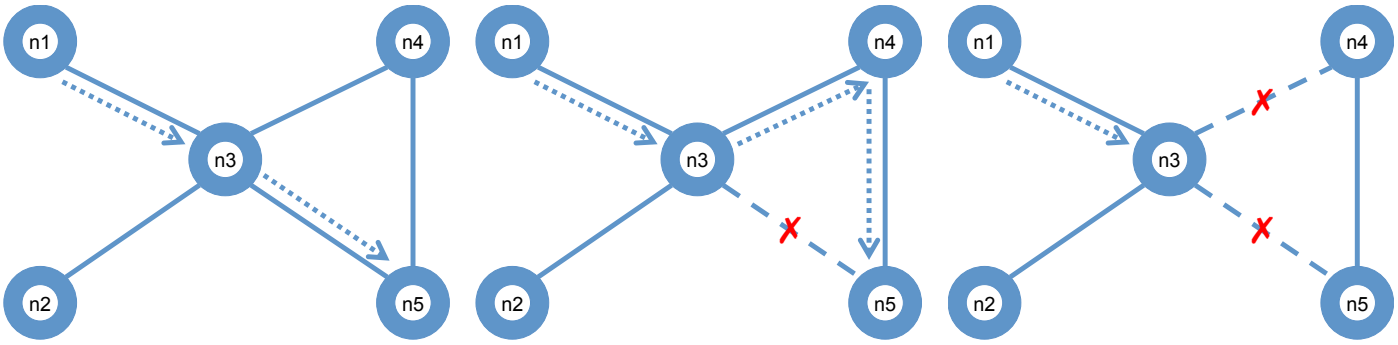
Fig. 9: Discretionary access control on forwarding nodes creates scope where some nodes become routers because they are provisioned to access both tag and interest.

III. Bleeding Edge Scenarios

We explore network topologies and node mobility patterns that enables certain networking constraints causing network to behave in unintentional way. We dwell our observations further by looking into security implementations in the system and the way these interacts with other layers. Based on these observations, we designs our experiments so that we can quantify the affect that securing the system brings.

A. Node trust affecting content routing

Provided all nodes act fairly and all edges have same brandwidth, shortest path between two nodes is the optimal path for communication. Illustrated in figure 11, if two nodes fail to co-certify, the path between those two nodes is virtually non-existent for any content exchange. This can lead to increase in latency as next optimal path can incur more hops. Furthermore, this phenomenon can also create components in the network graph. As there can be no communication



(a) Co-certification between two nodes in presence of one authority.

(b) Successfully co-certificating between two nodes depends on presence of mutually trusted authorities.

(c) Co-certification fails when two nodes can not show certificates signed by at least one mutual trusted authority.

Fig. 11: Node trust can increase latency. In worst cases, no data plane communication is possible with relatively isolated nodes.

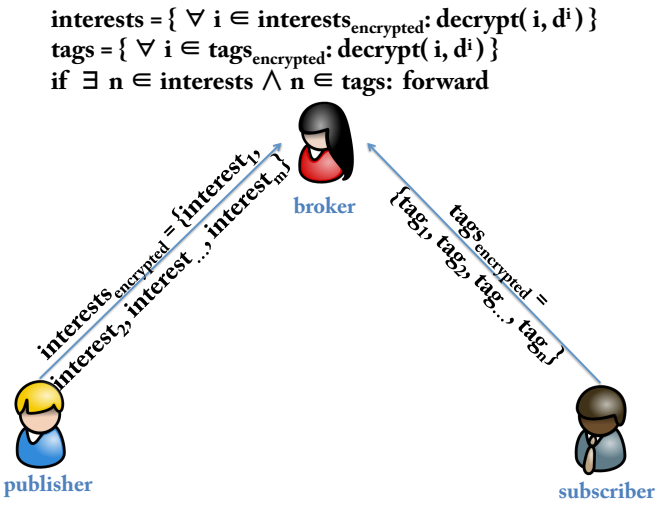


Fig. 10: Multiple tags and interests increases the data object availability to the network. It also increases the combined reach of scopes and forwarding ability of the network.

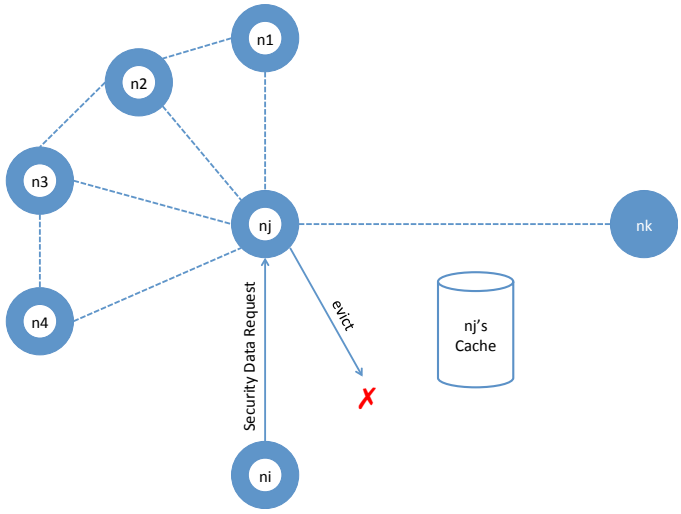


Fig. 12: Depending on caching policy, if node nj decides to evict a credential request data object from ni destined for authority nk, ni is unable to verify its authenticity in any further communications, obstructing its participation in data plane.

between graph components, it divides network and causes lack of communication between sets of nodes.

B. Security Data Objects effecting Network delivery

Data Objects generated in the network eventually hit network caches in the network. Depending of caching algorithms, these data objects can be replaced with existing data object in the cache. In certain cases, if a caching algorithm does not give preferential treatment to security data objects, it can result in e.g. as shown in figure 12, a node never getting its authentication credentials, hence not entering the pool of nodes that can exchange data objects.

C. Limited forwarding plane affecting on Content Delivery

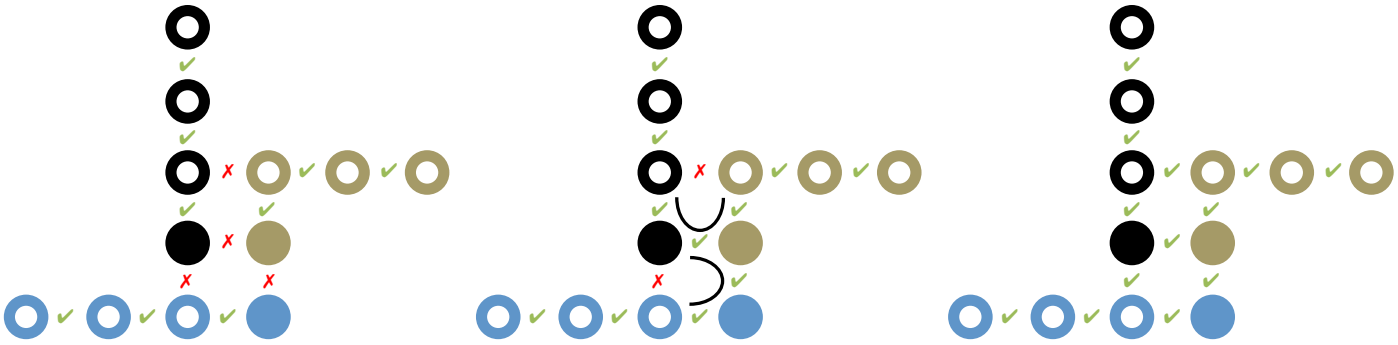
Setting access policies on tags and attributes will result in only provisioned node to decrypt them. In this case, a subset of nodes (brokers) can be fully authorized to decrypt both tags and attributes and hence forward data objects.

IV. Controlled Experiments

This section details experiments that we deployed to quantify the impact of security on different layers of ICEMAN stack. We devised these tests to highlight the effect of securing ICN by introducing node authenticity, content accessibility and privacy enforcement.

A. Methodology

1) Devices and Tools: We emulated real-world Android mobile performance on Ubuntu 12.04 (Linux). Using CPULimit tool, we matched ICEMAN daemon's performance on Android - by putting a constraint on ICEMAN daemons to use 30 percent of resources a ICEMAN daemon would use on Linux. Moreover, we used NRL's CORE as a container along-with EMANE for network emulation.



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

Fig. 13: Different co-certification scenarios. Solids are Authority nodes whereas circles are ordinary nodes. Only adjacent nodes are physically close enough to communicate directly.

B. Authenticating Publishers

As a secure ICN solution requires to authenticate publishing source, it is crucial to investigate how authenticating nodes can influence data object exchange patterns. We first draw a simple representative scenario and explore how trust among nodes can effect behavior of data plan.

1) Scenario: Figure 13 shows the physical arrangement of nodes in the experiment. Nodes are arranged in three group. Each group (collection of nodes with unique color in figure 13) has four nodes where one of the nodes is an authority node (shown as solid color). Nodes within a group trust each other as identity certificates for nodes in a group are distributed by the authority in that group.

2) Test construction: There are three cases which we took in consideration. (1) Only intra-group trust is established (as shown in figure 13a). This case deals with the situation where there can be no data object exchange between groups as no two nodes from different groups trust and co-certify each other. (2) Inter-group trust is established by letting all authority nodes in all groups co-certify each other. As shown in figure 13b, this allows communication between groups but a data object may not follow the shortest path to flow toward subscriber. (3) As shown in figure 13c, when security is not enabled, nodes no longer require to co-certify each other in order to exchange data object.

3) Data Object patterns: In the network, a total of 15 data object are published and 31 subscriptions are made. These publications and subscriptions are distributed among all groups. Theoretically total 348 data objects can be delivered in the network because many published data objects have same tags. All data objects are published and subscribed within first thirty seconds of network bootup whereas each data object had a payload of 512 KB.

4) Result: Network data object delivery is plotted against time in figure 14. Lack of inter-group trust halts any data object exchange across groups causing total networking delivery to reduce. A relatively lenient configuration allows inter-group communication but forcing data objects to follow longer paths causes slower network delivery. A universal trust environment or switching off security removes any barriers for data objects

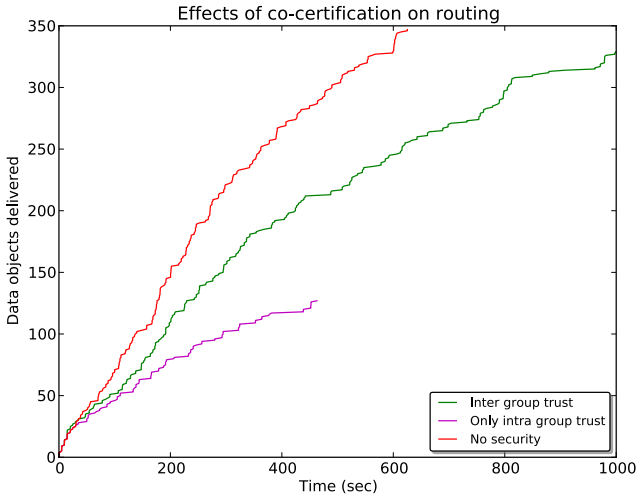


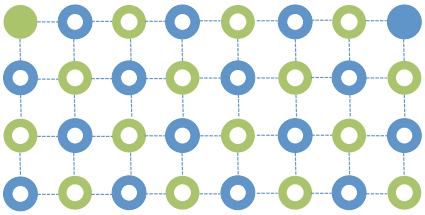
Fig. 14: Data object delivery patterns with different co-certification settings.

flowing between any two nodes. This cases full yet fastest data object delivery in the network.

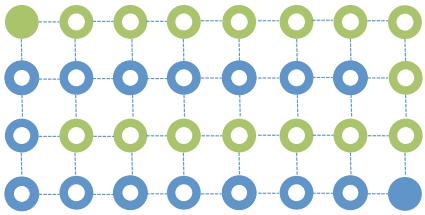
We conducted another detailed experiment geared to establish how effect of group formation can be reduced by enabling limited communication at certain points among groups.

1) Scenario: This experiment continuities the semantics developed in previous experiment. This experiment entitles two groups where nodes are equally distributed between these two groups. Each group has an authority that distributes identify certificates to all nodes in its own group.

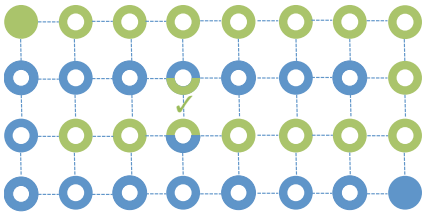
2) Test construction: We took following cases as they give a clear competitive formulation of topology: (1) Untrusted neighbors - Shown in figure 15a, we arrange nodes in a way that no two neighboring nodes co-certify each other. (2) Maximum sized linear sub-group - This setting (figure 15b) creates the longest path a data object can travel within a group. (3) Single bridge - Build on-top of case as shown in figure 15b, this case short-circuits the two groups in the middle - creating a bridge



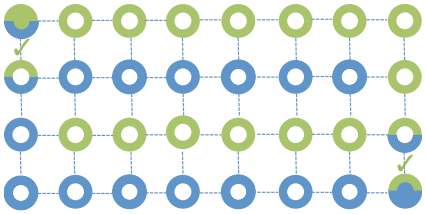
(a) Scenario 1: Untrusted neighbors.



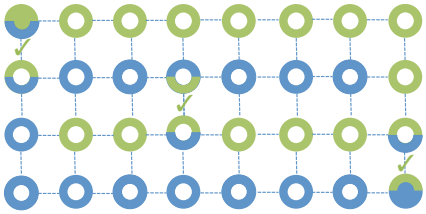
(b) Scenario 2: Maximum sized linear sub-group.



(c) Scenario 3: Single bridge.



(d) Scenario 4: Two bridges.



(e) Scenario 5: Three bridges.

Fig. 15: Different co-certification scenarios. Solids are Authority nodes whereas circles are ordinary nodes. Only adjacent nodes are physically close enough to communicate directly.

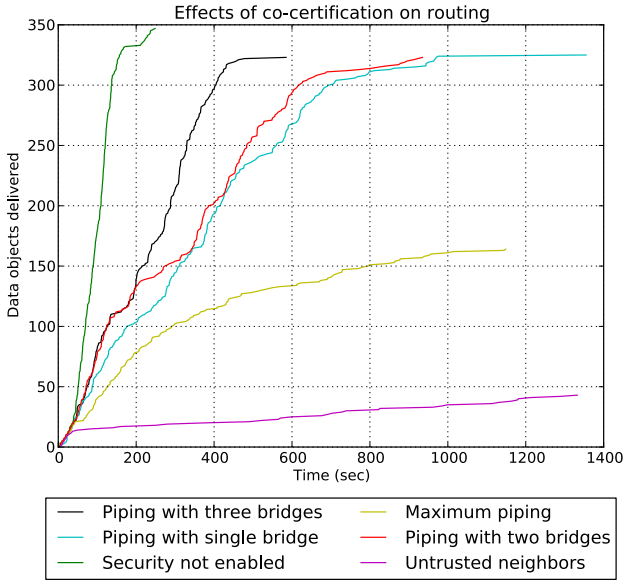


Fig. 16: Data object delivery patterns with different co-certification settings

between two groups. (4) Two bridges - Extending scenario illustrated in figure 15c, we create two bridges at the edges of two groups. This gives more room for data objects to travel across the groups. (5) Three bridges - This scenario fuses the communication room of single and double bridges cases. (figure 15c and figure 15d respectivery).

3) Data Object patterns: To remain consistent, we followed same data object generation and consumption pattern as followed in previous experiment.

4) Result: As show in figure 16, bridges have tremendous impact on speed of delivery of data objects. Increasing bridges not only creates outlets for data objects but also helps them find their destination using fewer hops.

C. Authorizing Subscribers

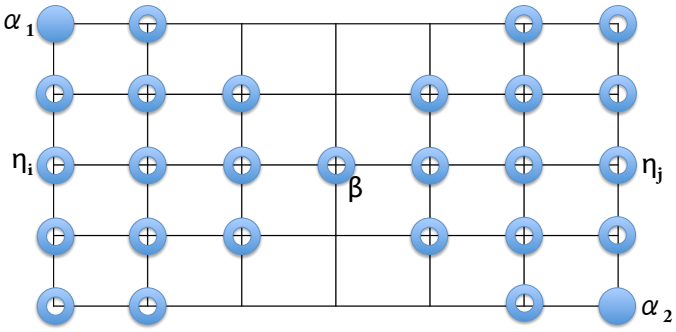


Fig. 17: Interaction of cache and security (fix it).

1) Scenario: We have a hourglass shaped network (figure 17) with two authorities α_1 and α_2 at each end of the network. These two sides of the network are connected by three linear nodes that forms the only physical connection between the two sides. Alongside signing of node certificates, these authorities also issue encryption and decryption attributes. A pair of encryption and decryption attributes is issued by same authority. Both authorities are configured to have same capabilities; both can certify and issue attributes to all nodes.

This creation of bottleneck has daunting effect on the caching and security data object exchange abilities of the network. A single data object delivery can result in multiple exchange of objects across the bridge. For instance, if a data

object, published by n_1 with an attribute from α_1 is subscribed by a node across the bridge, e.g. n_2, the request for decryption attributes will be sent across the bridge to α_1. A security data response with the decryption attributes from α_1 will also travel back from the bridge.

2) Test construction: We performed a comparative study of different caching strategies at different corners of the network. We developed tests around these cases: (1) Caching strategy at all nodes in the network prioritize security data objects over any other type of data objects. (2) All other nodes prioritize security data objects but bridge nodes do not differentiate between security data objects and other data objects. (3) Only bridge nodes prioritize security data objects. (4) Caching layers across the network are oblivious to the security data objects and treat them like any other data object.

3) Data Object patterns: We configured each node in the network to have a cache of 8 KB. In order to observe cache behavior, first we overwhelmed the cache at bridge node β. We achieved this by publishing and subscribing 8 data objects at β.

Each of 13 nodes on both sides of the network publishes one data object (total = 2 x 13). These data objects are encrypted with a unique attribute issued by the authority in the same side of topology as that of the node. Nodes on both sides make subscriptions in a way that each published data objects has to cross the bridge to get delivered on the other side of the network. Moreover, all subscriptions are equally distributed across all nodes to balance out the delivery overhead on individual nodes.

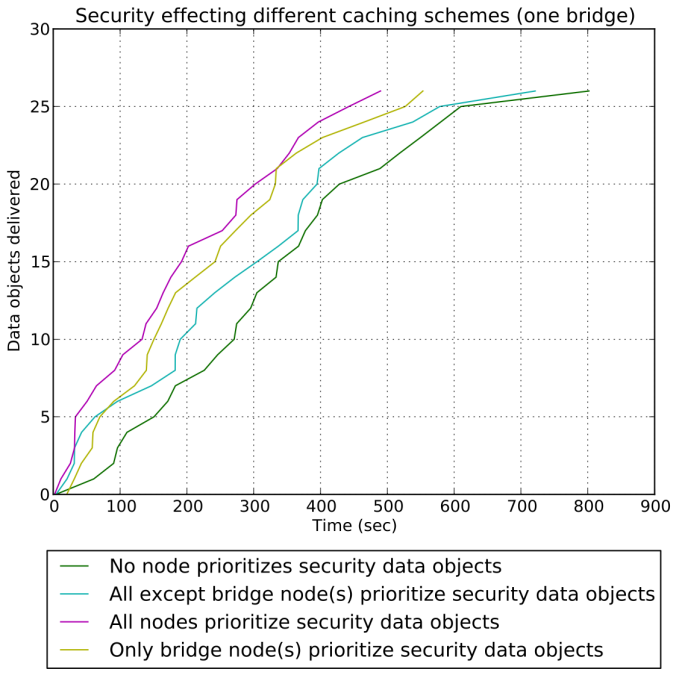


Fig. 18: In presence of security, data object delivery is effected by caching strategy.

4) Result: Time taken to deliver data objects are reported in figure 18. More nodes prioritizing security data objects leads to faster delivery. Prioritization at bridges has much more impact on rate of delivery than other nodes. Figure 20 shows variation in several test runs.

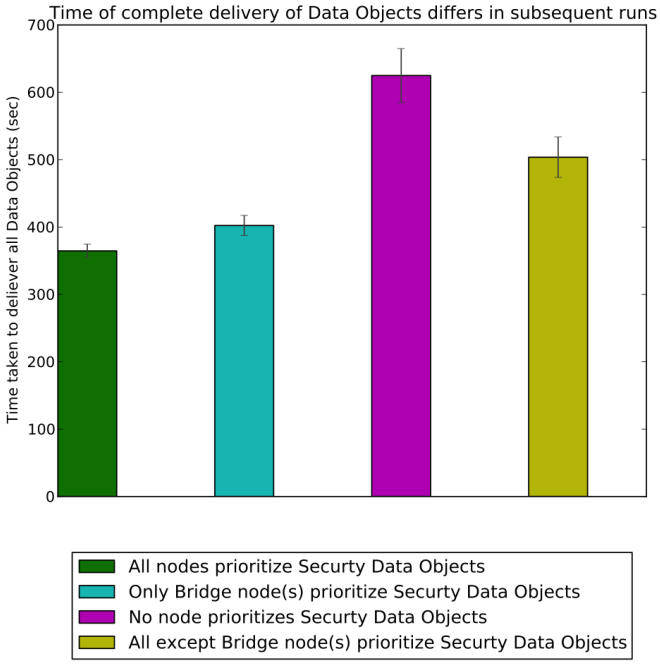


Fig. 19: Data object delivery pattern does not remain same during successive runs.

D. Selecting Brokers

1) Micro-Benchmarks: A router node can not perform routing if it is unable to decrypt tags and interests on incoming data objects or node descriptions. Therefore, all nodes that form a relying chain must perform decryption. We realized that decryption at each relaying node can be significantly time consuming task.

TABLE I: Performance benchmarks for security operations in ICEMAN.

Platform	ABE Attributes	Capability Generation	Encryption	Capability Usage	Decryption
Linux	1	35.653 (22.7 %)	1.210	20.857 (48.23 %)	0.9477
	2	57.909 (18.39 %)	1.215	38.654 (35.08 %)	1.207
	3	74.177 (14.32 %)	1.231	44.418 (46.35 %)	1.987
	4	100.455 (10.15 %)	1.601	46.503 (29.4 %)	2.1369
Android	1	251.781 (32.15 %)	7.916	129.526 (77.66 %)	6.334
	2	397.786 (26.77 %)	8.373	272.583 (49.75 %)	8.679
	3	554.821 (19.15 %)	9.002	302.975 (67.96 %)	15.349
	4	706.069 (5.94 %)	11.142	313.666 (43.59 %)	18.957

Capability generation and consumption in ICEMAN uses ABE encryption and decryption. These operations use consid-

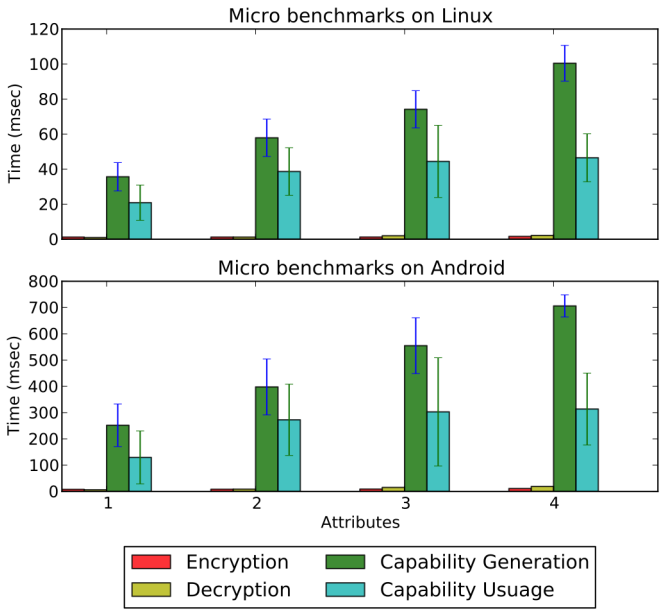


Fig. 20: Micro-benchmark variations over successive runs.

erable time (I) and time spend on individual nodes adds up and slows down network delivery.

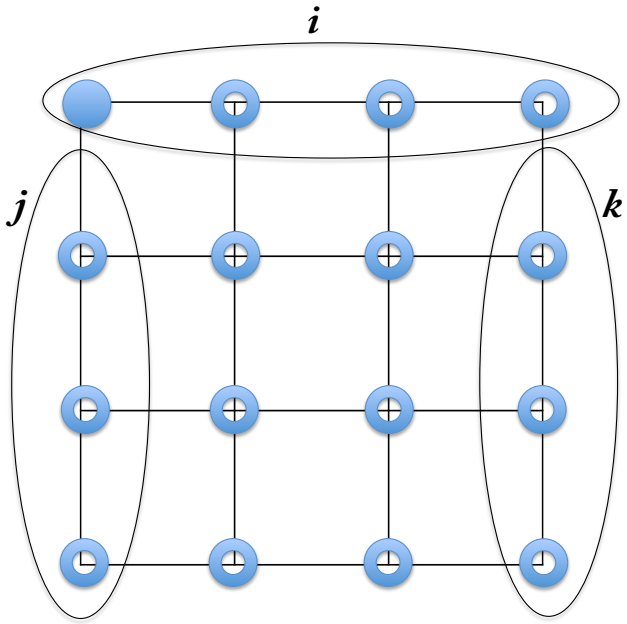


Fig. 21: 4 X 4 topology deployed to test effects of setting access policies on metadata.

2) Scenario: To further investigate this behavior, we designed an experiment where we can observe network effects of ABE decryption at individual nodes. Figure 21 shows a 4 X 4 grid of nodes where i nodes are publisher nodes where as i, j and k nodes are subscriber nodes.

3) Test construction: We ran three different types of tests on this topology: (1) Maximum linear sub-grouping where i nodes encrypt tags with a policy that only nodes in i, j and k can decrypt (and hence can relay). (2) Universal routing where all nodes have decryption attributes so all nodes can relay data objects and (3) When security is not enabled, not only all nodes can rely but the decryption overhead is also slashed at rely nodes.

4) Data Object patterns: Each publisher node publishers 4 data objects and each subscriber nodes subscribes to these data objects. Maximum network delivery is 160 data objects. These publications and subscriptions are made within first 20 seconds of network bootup.

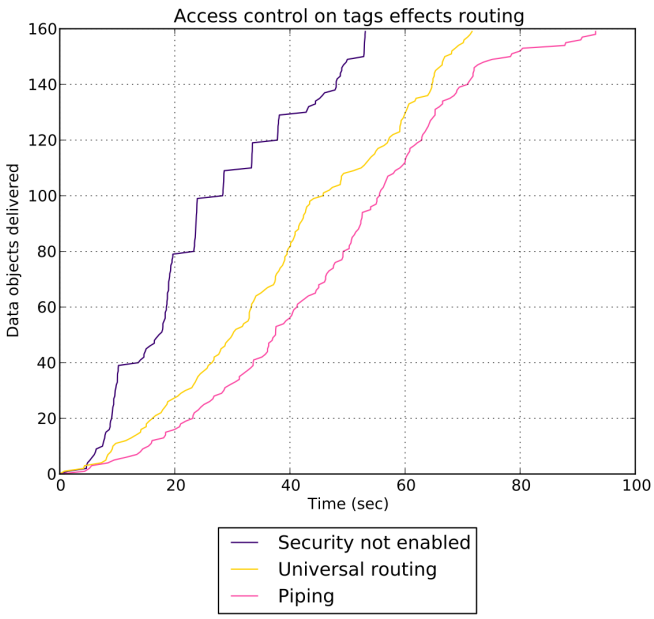


Fig. 22: Access control on tags and attributes slows down content delivery.

5) Result: A node can rely a data object if it has relative keys to decrypt meta data of subscribed interest and published data object. There is decryption overhead at each hop, causing considerable delay. This delay is reduced if there can be multiple paths between two nodes as nodes in different paths 'compete' to forward a data object faster. Data Object delivery time is smallest when security considerations are not taken in place. This is illustrated in figure 22.

V. Analysis

VI. Conclusion

References

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

[2] Dan Boneh, Giovanni Di Crescenzo, Rafail Ostrovsky, and Giuseppe Persiano, Public key encryption with keyword search, *23rd International Conference on the Theory and Applications of Cryptographic Techniques*, Springer, 2004.

[3] Dan Boneh, Amit Sahai, and Brent Waters, Functional encryption: Definitions and challenges, *8th Theory of Cryptography Conference*, Springer, 2011.

[4] Reza Curtmola, Juan Garay, Seny Kamara, and Rafail Ostrovsky, Searchable symmetric encryption: Improved definitions and efficient constructions, *13th ACM Conference on Computer and Communications Security*, 2006.

[5] Nikos Fotiou, Giannis Marias, and George Polyzos, Access control enforcement delegation for information-centric networking architectures, *2nd ACM Workshop on Information-Centric Networking*, 2012.

[6] Sanjam Garg, Craig Gentry, Shai Halevi, Mariana Raykova, Amit Sahai, and Brent Waters, Candidate indistinguishability obfuscation and functional encryption for all circuits, *54th IEEE Symposium on Foundations of Computer Science*, 2013.

[7] Ali Ghodsi, Teemu Koponen, Jarno Rajahalme, Pasi Sarolahti, and Scott Shenker, Naming in content-oriented architectures, *1st ACM Workshop on Information-Centric Networking*, 2011.

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

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

[10] Ronald Rivest, Adi Shamir, Leonard Adleman, A method for obtaining digital signatures and public-key cryptosystems, *Communications of the ACM*, Vol. 21(2), 1978.

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

[12] Van Jacobson, Diana Smetters, James Thornton, Michael Plass, Nicholas Briggs, and Rebecca Braynard, Networking named content, *5th International Conference on Emerging Networking Experiments and Technologies*, 2009.

[13] 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.

[14] George Xylomenos, Christopher Ververidis, Vasilios Siris, Nikos Fotiou, Christos Tsilopoulos, Xenofon Vasilakos, Konstantinos Katsaros, and George Polyzos, A survey of information-centric networking research, *IEEE Communications Surveys and Tutorials*, Vol. 16(2), 2014.

[15] @inproceedings39294118, author = Soon-Young Oh and Davide Lau and Mario Gerla, title = Content Centric Networking in tactical and emergency MANETs, booktitle = Wireless Days, WD, year = 2010, pages = 1–5, doi = 10.1109/WD.2010.5657708, masid = 39294118