Secure Producer Mobility in Information-Centric Network

Alberto Compagno, Xuan Zeng, Luca Muscariello, Giovanna Carofiglio, Jordan Auge

Secure Producer Mobility in Information-Centric Network

Alberto Compagno*, Xuan Zeng†, Luca Muscariello*, Giovanna Carofiglio*, Jordan Augé*

* Cisco Systems † IRT SystemX, UPMC

ABSTRACT

One of the fundamental requirements of the next generation 5G networks is to support seamless mobility over an heterogeneous access network by design. The shift from host-based to content-based location-independent communication makes Information-Centric Networking (ICN) an appealing technology to provide not only mobility, but also security and storage as native properties of the network architecture.

Previous work in ICN literature focused on name-based mobility management solutions and particularly on the challenges of producer mobility, which involves an interaction between forwarding and control plane.

In this paper, we consider the security implications of producer mobility in ICN and we highlight the importance of securing producer to network interactions. We focus on the problem of *prefix hijacking*: a class of attacks that can be exploited to threaten both the security of the ICN networks and the privacy of its users. To prevent this class of attacks, we propose a fully distributed and very low-overhead protocol for *name prefix attestation* based on hash-chaining. First results show order of magnitudes improvement in verification latency with respect to signature verification, the leading alternative approach to thwart prefix hijacking attacks.

CCS CONCEPTS

• **Security and privacy** → **Security protocols**; **Mobile and wireless security**; • **Networks** → *Mobile networks*;

KEYWORDS

ICN Security; Producer Mobility; Prefix hijacking

1 INTRODUCTION

The upcoming 5G technology is expected to provide ubiquitous Internet access and seamless user mobility over a denser and increasingly heterogeneous network access, integrating multiple radios (e.g., LTE and Wifi in addition to 5G radio). To strengthen the importance of an effective, fast and low-overhead mobility management, 5G networks will accommodate more stringent requirements in terms of lower latency, higher throughput, stronger reliability and availability to offer improved experience and new services to the users (e.g., 4K video streaming, VR/AR).

Information-Centric Networking [32](ICN) is emerging as a promising networking paradigm to meet 5G requirements due to its native support for mobility, storage and security at network layer. ICN communication adheres to the consumer-producer model: producers generate and publish under a specific name prefix (e.g., `cisco/cicn`) content that consumers can obtain by sending named requests (called Interests). Interests are forwarded in a name-based fashion up to the producer to retrieve corresponding content (e.g., `cisco/cicn/sb-forwarder.apk`). Optionally, intermediate routers can serve content directly by their cache.

Two design principles enable native support of consumer mobility in ICN: (1) content names are location-independent, (2) communication is consumer(rather than producer)-driven. Supporting producer mobility is instead more challenging in ICN. Every time a producer moves, the network must adjust its forwarding information to maintain producer reachability, i.e., interests matching the producer's prefix must be forwarded to new producer's location.

Several proposals for handling producer mobility exist in ICN literature (cfr. [33] for a survey). Among them, tracing-based protocols [5, 17, 21, 30, 34] exploit the ICN stateful forwarding plane to minimize the communication delay due to mobility, packet losses and signaling overhead. In order to preserve producer's reachability, tracing-based protocols entitle the producer to update the network forwarding plane after every mobility event. To this aim, a producer issues a special interest, we call it *Interest Update*, that leaves a breadcrumb in the forwarding state of the traversed routers. Regular interests reach the new location of the producer by following the breadcrumb trail left by interest updates.

Deploying tracing-based protocols without adequate security mechanisms pose serious security threats for both the network and the producers. More precisely, a producer should be able to issue legitimate Interest Updates only for the prefix(es) it is entitled to use for publication. If there are no security mechanisms to enforce such rule, an attacker can easily forge Interest Updates for other producers' prefix(es) and thus divert consumers' requests to him (i.e., *prefix hijacking attack* [7]). By doing so, the attacker can: perform black-hole attacks to its victims [3], make genuine content cached in the network unreachable and pollute in-network caches with bogus content [12], prevent consumers from receiving the content they asked for [16], collect consumers' interests to attack their privacy [4].

To tackle such security concerns, in this paper we introduce a fast and lightweight *prefix attestation protocol* that gives to a producer the ability to express genuine Interest Updates only for its prefix(es). Our protocol is designed to run unchanged on different hardware deployed at network access (e.g., micro, nano, small 4G/5G cells as well as Wifi access points) and at mobile core network.

We compare our approach with a signature based mechanism adopted in most of the prefix attestation proposals [9, 19, 20, 28, 31, 34]. Results show that our lightweight approach maintains more than 90% of the original goodput (i.e., without any security mechanism), while the signature approach drops it close to 0%. In terms of storage requirements, our protocol only requires tens of megabyte on each router to manage billions of mobile producers.

The rest of the paper is organized as follows: Section 2 discusses related work before introducing the design of our protocol in Section 3. Section 4 presents a detailed description of our proposal.

Section 5 discusses the security of our protocol and Section 6 evaluates its performance. Section 7 concludes the paper.

2 RELATED WORK

Among the mobility management protocols proposed for ICN [5, 6, 18, 21, 23, 24, 30, 34], Kite [34] is the only one that takes into account security in its design and protects the network against prefix hijacking. Specifically, authors propose to sign traced interests (which corresponds to our Interest Updates) using the producer's private key, in order to handle mobility in a secure fashion. Every router receiving a traced interest verifies the signature before updating its network state. The binding between the producer's public key and the producer's prefix(es) (e.g. through certificate) attests producer's entitlement to generate traced interest. However, this approach has some drawbacks: routers must be aware of the producers' certificates, signature verification and certificate chain traversal increases latency during handover, revoking of a prefix to a producer faces the same problem of certificate revocation [26].

Few works have proposed prefix attestation mechanisms[1] to prevent prefix hijacking in mobility protocols for IP networks. Both Cellular IP [10] and TeleMIP [14] adopt the following approach: the first time a host connects to a network gateway, the gateway assigns an address, a host id and a session key to the host. During handover, the host uses the session key to authenticate itself and prove the ownership of the IP address to the new access point. The main drawback of such approach is the use of a single network key to generate every host's session key. In case the network key is stolen, e.g., when a router is compromised, a new network key and a refresh of all session keys must be performed.

Prefix attestation has also been proposed to prevent IP prefix hijacking in inter-domain [9, 19, 20, 28, 31] and intra-domain [25, 29] IP routing. A widely used approach for achieving address attestation exploits digital signatures and certificates. A trusted address holder issues a singed certificate that attests the router's right of announcing a specific address prefix in the network. Both sBGP [20] and soBGP [31] use a public key infrastructure to establish trust between address holder and BGP routers. Similarly, authors of [25] propose to use signed certificates to attest the list of network prefixes an OSPF router can announce to different OSPF areas (i.e., through Router Links LSA messages). While the same approach can be applied to the tracing-based mobility protocols, it would suffer of the same issues we discussed for Kite.

Finally, two different approaches for address attestation are proposed in psBGP [28] and s-RIP [29]. psBGP proposes a decentralized mechanism: each AS creates a prefix assertion list (PAL), that contains address ownership assertions of the local AS-es and its peers. An origin claim is validated by checking the consistency between the PALs of peers around the advertising origin. S-RIP [29] achieves prefix attestation pre-distributing the mapping between router ids and prefixes to announce in every router of the network. Both mechanisms work well when the mapping between router and prefixes is almost stable. It is worth noticing that, instead, in tracing-based mobility protocols for ICN such mapping may vary very frequently.

3 PREFIX ATTESTATION PROTOCOL DESIGN

We design our protocol on top of the proposed tracing-based mobility protocols [5, 34], extending them by: (i) introducing a *bootstrap phase* that authenticates a mobile producer when it first connects to the network; (ii) adding a *Secure Interest Update Validation* mechanism. Instead, we leave the underlying tracing-based mobility protocol to decide when and how to stop propagating Interest Updates.

The bootstrap phase will authenticate the producer to the network, giving evidence of its entitlement to announce its prefix(es). We want to highlight that our producer authentication is different in principle from the user authentication employed in many mobile networks (e.g., 3G/4G and Wifi). We recall that the latter is used to allow (or deny) a user to connect a device to a mobile network and it does not give any insight of what a device can publish. Moreover, we believe that user identities and producer identities should be managed separately. A user might authenticate different devices to the network using its user identity[2], while such devices might need to announce different sets of prefixes due to the producer applications they run[3]. Authenticating user identities and producer identities separately gives the needed flexibility to cover the aforementioned case.

The secure Interest Update validation mechanism guarantees that only the entitled producer can generate a legitimate Interest Update for the prefix(es) under its responsibility. Moreover, it allows each router of the network to verify the validity and freshness of the Interest Update through attestation: every Interest Update carries a fresh proof that it has been generated by the entitled producer.

In the following, we present the system model and the threat model of our proposal.

3.1 System Model

Our protocol involves four types of network entities: registration server, core router, edge router and mobile producer. Figure 1 depicts the system model considered throughout the paper.

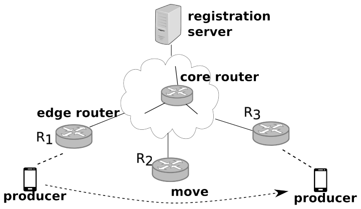


Figure 1: System Architecture

The registration server is mainly responsible for: (1) authenticating the mobile producer and verifying the ownership of the prefix(es) it announces, (2) generating and distributing to the network the necessary cryptographic material to validate Interest Updates for the producer's prefix(es). We call **security context** such amount of cryptographic material, and we assume that the **security context** of a given prefix can be stored as an additional field

[1] In IP, such mechanisms are usually called address attestation. We maintain the name "prefix attestation" for ease of exposition.

[2] As in the password-based authentication mechanism adopted in many Wifi networks.

[3] Consider a user owning a mobile phone and a laptop. The mobile phone runs an VoICN application generating content for the prefix \A, while the laptop does not run the same application. The laptop should not be entitled to announce the prefix \A until the VoICN application is installed. If this rule is not guaranteed, a malware running on the laptop might announce prefix \A and perform the attacks described in [4, 12, 16].

in the forwarding state of the same prefix (e.g., PIT or FIB)[4]. We define a mobile producer as a mobile device storing a producer identity entitled to publish content under one or more prefixes. A pair of private/public keys is associated to the producer identity and used to sign/verify the content it publishes. Finally, we consider the network to be composed of edge and core routers forming a single Autonomous System.

We consider the access of the network to be heterogeneous (e.g., edge routers can be 4G/5G cell or WiFi access points). Moreover, every mobile device knows valid user credentials to connect to the network infrastructure. Once the authentication is performed, the communication between the mobile device and the edge router is considered secure (i.e., encrypted and integrity protected).

3.2 Threat Model

In this work, we consider an attacker that controls mobile devices that can connect to the network, e.g., the attacker buys a valid SIM. The attacker targets genuine mobile producers and aims at generating legitimate Interest Updates for the prefixes used by its victims.

We assume that edge routers can be compromised by the adversary, while core routers and the registration server are trusted. These assumptions are consistent with the assumptions made on the existing mobile networks [11]. Moreover, we assume an intrusion detection mechanism is in place and it is able to detect a compromised edge router [8]. As soon as a compromised edge router is detected, it is disconnected from the network along with the devices connected through it. Finally, we assume that the attacker can access every information stored in the compromised edge router.

4 PREFIX ATTESTATION

Our proposal exploits route versioning to verify that an Interest Update is fresh and not the result of a replay attack[5]. In particular, for a given prefix a router stores a sequence number in the corresponding forwarding state of such prefix. A router considers an Interest Update fresh only if the interest carries a greater sequence number than the one stored in the router.

We make use of **one-way hash-chain mechanism** to guarantee that a producer can generate Interest Updates only for its own prefix(es). One-way hash-chain is a simple mechanism initially proposed by Lamport [22] as a replacement for password-based user authentication and authorization (e.g., a user A that wants to log-in to a server B for accessing its service). The mechanism works as follows: user A generates a sequence of values by applying n times a cryptographic hash function H to a random value s as depicted in Figure 2. The value s is called *root of the chain.*

We assume that initially B receives $H^n(s)$ and is assured of it genuineness. When the user A wants to log-in on B it sends $H^{(n-1)}(s)$ to B. B simply checks that $H(H^{(n-1)}(s)) = H^n(s)$. This proves that only A could have generated $H^{(n-1)}(s)$.

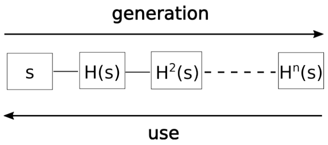


Figure 2: Hash Chain illustration.

We use the hash-chain mechanism in the following way. We associate an hash-chain for each producer's prefix such that: $H^{(n-i)}(s_p)$ is the hash value corresponding to the forwarding state for the prefix p with sequence number i. Therefore, we consider the **security context** for a prefix p to be a value of the chain $H^{(n-i)}(s_p)$ stored in a router, such that $i < n$. A router considers an Interest Update for a prefix p valid only if: (1) the Interest Update carries a hash value $H^{(n-j)}(s_p)$ where j is the corresponding sequence number; (2) the security context for p in the router is $H^{(n-i)}(s_p)$ and $j > i$; (3) the equality $H^{(j-i)}(H^{(n-j)}(s)) = H^{(n-i)}(s_p)$ holds. Table 1 reports the notation we use throughout the paper.

RS	Registration Server
MP	Mobile Producer
ER	Edge router to which **MP** is connected
p	Prefix owned by **MP**
s_p	Root of the hash-chain for prefix p
n	Length of the hash chain of p
H	Cryptographic hash function
$H^{(n-i)}(s)$	Value of the hash chain for the sequence number i

Table 1: Notation table.

In the following sections, we describe in more details the bootstrap phase and the secure Interest Update validation mechanism we introduced in Section 3.

4.1 Bootstrap phase

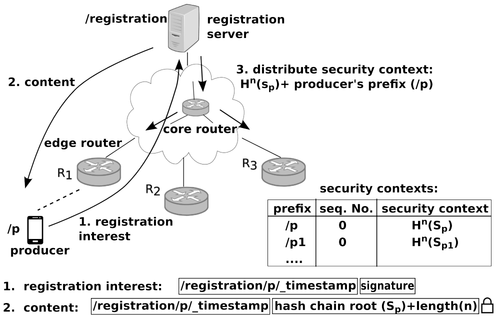
The bootstrap phase is illustrated in Figure 3. The bootstrap starts with **MP** authenticating and proving to **RS** its right of announcing its prefix p. To achieve this, **MP** issues an interest to **RS** containing p as a name component. Such interest, we call it registration interest, will be signed with the **MP**'s private key and it will carry a fresh timestamp. Once **RS** has verified the signature and checked the freshness of the registration interest, **RS** sends back to **MP** a content containing s_p and n, the root and the length of the hash chain for p. The content payload will be encrypted with **MP**'s public key so that only **MP** can access the root of the chain. Then, as shown in Figure 3, **RS** will distribute the initial p's security context (i.e., the hash value corresponding to the sequence number 0) to the whole network[6]. We envision that the distribution of the security context can be performed using a routing update message.

The protocol ends with **MP** generating the full hash chain and issuing an Interest Update with sequence number 1 to the edge router to which it is connected. This Interest Update is important to prevent a prefix hijacking attack in this step, as will be explained later in Section 5.1.

[4] We highlight that our protocol can be adopted in any ICN architecture that stores the forwarding state (i.e., NDN, MobilityFist, XIA). We refer to the PIT and FIB as an illustration, without limiting the applicability of our protocol to any specific ICN architecture.

[5] Our route versioning can be used together with the one adopted in [5] or it can be considered an additional mechanism.

[6] Since Kite does not permanently store the forwarding state for each prefix in every router, routers that do not have any forwarding state p will drop the security context


/registration
registration server
2. content
3. distribute security context: $H^n(s_p)$+ producer's prefix (/p)
core router
edge router
R_1
R_2
R_3
/p
producer
1. registration interest
security contexts:

prefix	seq. No.	security context
/p	0	$H^n(S_p)$
/p1	0	$H^n(S_{p1})$
....		

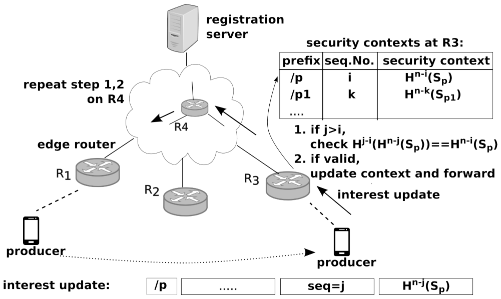
1. registration interest: /registration/p/_timestamp signature
2. content: /registration/p/_timestamp hash chain root (S_p)+length(n)

Figure 3: Bootstrap phase

4.2 Secure Interest Update Validation

Figure 4 shows our Secure Interest Update Validation mechanism. According to the tracing-base protocols, at every mobility event (i.e., when **MP** connects to a different **ER**) **MP** issues a new Interest Update. Our validation mechanism requires that the Interest Update carries a sequence number, monotonically incremented at every release, and the proof of validity. In the following, we describe our mechanism assuming **MP** releases an Interest Update with a newer sequence number j, and the corresponding value of the hash-chain $H^{(n-j)}(s)$ as proof of validity. We only describe the verification steps performed at **ER**, although it has to be noted that every router, core or edge, will perform the same verification steps when receiving an Interest Update.

Upon reception of the Interest Update, **ER** matches the name to its forwarding state and retrieves the related security context $H^{(n-i)}(s)$, together with the corresponding sequence number i. Then, **ER** verifies that $j > i$ and, if the inequality holds, it compares $H^{(j-i)}(H^{(n-j)}(s))$ to $H^{(n-i)}(s)$. If the two values match, **ER** updates the corresponding forwarding state and the security context to $H^{(n-j)}(s)$. Finally, **ER** sends the Interest Update to the next (edge or core) router according to the mobility management protocol in place[7]. Figure 4 shows the verification step performed by **ER**.


registration server
security contexts at R3:
repeat step 1,2 on R4
R4
edge router
R_1
R_2
R_3
1. if j>i, check $H^{j-i}(H^{n-j}(S_p))==H^{n-i}(S_p)$
2. if valid, update context and forward
interest update
producer
producer

prefix	seq.No.	security context
/p	i	$H^{n-i}(S_p)$
/p1	k	$H^{n-k}(S_{p1})$
....		

interest update: /p ..... seq=j $H^{n-j}(S_p)$

Figure 4: Secure Interest Update Validation

Remark. It is worth noticing that the verification steps described so far do not fully prevent prefix hijacking attacks. In particular, there is no guarantee that every router in network has the most recent version of the forwarding state. Therefore, outdated routers could not be able to detect old or replayed Interest Updates, if the Interest Updates carry a greater sequence number than the router's security context.

To solve this problem, we exploit a common property of all the tracing-base protocols that is: an Interest Update always hits a router with the most recent version of the forwarding state[8]. Therefore, if the check $j > i$ fails for a certain Interest Update, it means that such Interest Update is old (i.e., it carries an old sequence number) and all the upstream routers that accepted such old Interest Update have a corrupted forwarding state. To restore the corrupted forwarding state, a router drops any old Interest Update it receives, and creates a new Interest Update using its local security context. This new Interest Update is then propagated back to the upstream routers, updating and fixing their forwarding state.

5 SECURITY CONSIDERATIONS

In the following sections, we provide a security discussion regarding the threat model presented in Section 3.1. We show how an attacker will not be able to successfully conclude neither the bootstrap nor the secure interest validation mechanism for a prefix p that does not belong to it. Moreover, we discuss about a possible denial of service attack that might exploit the design of our protocol. For it, we also propose a mitigation mechanism.

5.1 Preventing Prefix Hijacking attacks

We consider the case in which the attacker tries to pass the registration phase in order to deploy a security context for the prefix p. To successfully pass the registration phase, the attacker needs to generate a valid signature for the registration interest for p. The signature must be calculated with the private key of the producer that owns p. However, since the private key is secretly store in the producer's device(s) and never transmitted to the network, the attacker cannot generate a new valid registration interest by its own. Its only chance to pass this step is to replay a valid registration interest. We recall that an attacker can compromise an edge router, and by reading the edge router's memory, it can obtain a valid registration interest. If such interest has already been received by **RS**, the attacker will not be able to pass the registration phase (the timestamp in the registration interest will reveal the replay attack). If the registration interest has not been received by **RS** (e.g., due to congestion), the registration interest will be accepted. At this point, the attacker must be able to express a valid Interest Update to get the edge router to update its forwarding state. However, because the attacker does not know the producer's private key, it will not be able to decrypt the content received by **RS** and access the root of the chain for p. In the following, we argue that an attacker cannot generate a valid Interest Update without knowing the root of the chain, thus neither conclude the bootstrap phase nor pass the secure Interest Update validation mechanism.

To express a valid Interest Update for a prefix p, an attacker must be able to generate $H^{(n-i)}(s)$ such that $i > j$ and $H^{(n-j)}(s)$ is the latest value of the chain released by the genuine producer. In our design, we assume that the hash-chain is generated using a cryptographic hash function (e.g. SHA256). The security properties of cryptographic hash functions (i.e., Pre-image resistance, Second pre-image resistance and Collision resistance) make impossible to generate $H^{(n-i)}(s)$ without knowing any $H^{(n-k)}(s)$ where $k >$

[7] In Kite, a router might not have the security context for p. In this case we exploit the acknowledge message to let the anchor to send the corresponding security context. Interest Update validity will be check after receiving the security context.

[8] For Map-Me protocol this is formally proven. For Kite, an Interest Update always arrives to the anchor. The anchor the will always have the most updated security context

i [27]. Because a producer releases the values of the chain in the reverse order, it is easy to prove that either the attacker knows the root of the chain or it cannot generate any $H^{(n-i)}(s)$.

5.2 Denial of Service Attacks

In this section, we discuss how the validity check of Interest Updates might open a door to Denial-of-Service attacks to the edge routers. We consider the case in which an attacker issues a non-legitimate Interest Update for p that hits a router r. Such non-legitimate Interest Update carries a sequence number i such that $i \gg j$, and j is the sequence number of the forwarding state for p in r. To be able to detect the Interest Update as non-legitimate, r needs to hash $i - j$ times the security context associated with p. The greater is the distance between i and j, more hashes r will need to calculate. An attacker can issue non-legitimate Interest Updates with great sequence numbers to keep the router busy on calculating hashes, thus provoking a DoS attack to the other connected producers.

To prevent the above DoS attack, we set a threshold t so that every router will drop Interest Updates whose sequence number i is greater than $t + j$. However, using a threshold-based approach brings another problem. A router with an old version j might drop valid Interest Updates because the sequence number i they carry is greater than $t + j$. This might happen if a mobile producer moves frequently between a small subset of edge routers. To avoid this problem, we propose to exploit a routing protocol to maintain the security context of the routers loosely synchronize. Every routing update message will carry the security context of the router generating it, along with the regular routing information. We leave for future work the full design of the mechanism and the evaluation of the overhead introduced by it.

6 EVALUATION

In this section, we evaluate the overhead introduced by our protocol in both routers and mobile producers. We focus our evaluation on the Secure Interest Validation mechanism because it is the most frequent (i.e., at every mobility event) and the most demanding step (i.e., it requires cryptographic operations) of our protocol. In particular, we provide an analysis of (i) computational overhead and (ii) additional storage cost involved in the routers. We leave for future work a full evaluation of the protocol including the bootstrap phase.

We compare our hash-based verification with the signature-based verification adopted in most of the prefix attestation proposals [9]. We consider the protocol to be the same in both approaches (i.e., both issue an Interest Update that will be verified at each router). In the hash-based verification an Interest Update will carry a hash value while, in the signature-based approach, the Interest Update will be signed with the producer's private key. Results show that our approach reduces both computational and storage overhead. In particular, the lower computation overhead of our mechanism allows a router to maintain more than 90% of the original goodput (i.e., with no verification) while the signature approach drops it close to 0%. Moreover, our approach reduces by 66% the storage overhead introduced by the more expensive signature based approach.

6.1 Computational Overhead

To evaluate the computational overhead, we quantify the time required to perform a verification with both hash-based and signature-based approaches. Then, based on a simple analytical model, we derive their impact on the overall router goodput (i.e., the number of regular packet processed by the router excluding Interest Updates) as the producer mobility rate increases.

The time required to verify an Interest Update can be characterized as the sum of the time needed to retrieve the security context for the Interest Update and the time to verify the proof in the Interest Update (i.e., either the hash or the signature). Retrieving the security context only adds a negligible time. In fact, the security context is stored together with the forwarding state in the corresponding table and it can be retrieve during the regular lookup for processing an interest[9]. Therefore, the time needed to verify the proof is the dominating factor in both the hash-based verification or the signature-based verification.

We evaluate the two verification mechanisms considering the hardware adopted on the current edge routers. We choose the edge routers over the core routers for our evaluation because the former are less capable than latter, and so more sensitive to the computational overhead. We consider the Cavium Octeon MIPS64 as the reference hardware[10] and we get both hash calculation time and signature verification time from the openwrt [2] benchmark for MIPS64 processors [1]. Table 2 reports the time required for the hash-based verification and the signature-based verification.

Hash-chain based		Signature based	
SHA256	MD5	RSA	DSA
$3\mu s$	$0.8\mu s$	$4700\mu s$	$5710\mu s$

Table 2: Verification delay.

From Table 2, we can observe that computational overhead incurred by the hash-based verification is about three orders of magnitude smaller than the computational overhead incurred by the signature-based verification. It is interesting to note how a single hash can be calculate in a fraction of micro seconds, meaning that a router can apply a hash function to a packet and still process such packet at line rate. We believe this is important to prevent Denial Of Service attacks that exploit the computational overhead. The signature verification cannot be done at line rate and it can thus open the door to Denial Of Service attacks.

Then, we investigate the impact of the verification delay on edge router's goodput increasing producer's mobility rate. We calculate the edge router's packet goodput from the model presented in [15]. We consider η to be the fraction of Interest Updates over the total number of packets received by a router, the goodput(in packet/s) can be calculated as:

$$goodput = \frac{1 - \eta}{\tau_{process} + \eta \times \tau_{verif}} \qquad (1)$$

where $\tau_{process}$ is the average processing time for a normal packet, τ_{verif} is the verification delay for an Interest Update message. For the edge router, we consider a maximum throughput of

[9] For the signature verification, the producers' public keys are the security context.
[10] Cavium is leader in carrier-grade wireless access routers (LTE and/or WiFi)

0.25Mpps, thus $\tau_{process}$= 4μs. For τ_{verif} we apply the number reported in table 2.

From equation 1, we can compute the edge router's goodput in packet/s. Figure 5 shows how the edge router's goodput decreases as we increase the amount of producer mobility message.

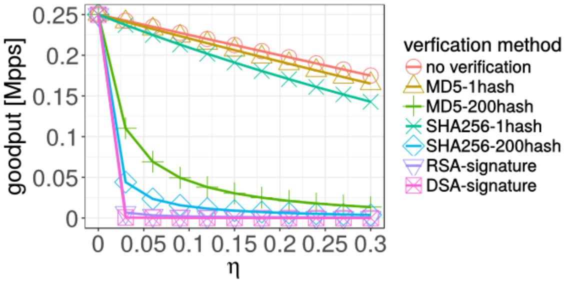


Figure 5: edge router goodput

With the signature-based approach, the goodput decreases to 0 Mpps with only about 5% of Interest Update. With the same amount of Interest Update, our hash-based mechanism can maintain the 95%-98% of the original goodput (i.e., no verification performed) when one hash computation is required per Interest Update. Overall, our mechanism achieves 80%, with the slower SHA256, and more than 95%-98%, with the faster MD5, of the original goodput with one hash computation per Interest Update. Figure 5 also shows that with around 200 hashes computation per Interest Update our approach shows comparable performance to the signature-based approach. Therefore, maintaining updated the context state in the routers allows our mechanism to perform the best and to achieve a substantial gain when compared with the signature based approach.

We assume that the producer stores the full chain at the bootstrap phase. Therefore, during any handover it will not need to do any hash computation. Although this approach minimizes the computational cost for issuing Interest Updates, it increases the computational overhead at bootstrap. To reduce the computational overhead of the hash-chain calculation we can adopt the mechanism proposed by Coppersmith and al. [13]. Such mechanism provides a computation complexity of $\frac{1}{2}log_2 n$ for calculating the full chain.

6.2 Additional Storage Cost

Every edge and core router needs to maintain a security context for each of its forwarding state. Since our mechanism stores the security context in the same structure containing the forwarding state (e.g., PIT or FIB), the storage cost can be calculated as:

$$Storage\_cost = N_{forwarding\_entry} \times (S_{security\_context} + S_{seq}) \quad (2)$$

where $N_{forwarding\_entry}$ is the number of entries in the forwarding structure, $S_{security\_context}$ is the size of the crypto material needed to perform the verification(i.e., either a hash or a public key), S_{seq} is the sequence number corresponding to the forwarding state version. For the hash-based mechanism we assume that $S_{security\_context}$ = 32 bytes while for the signature-based mechanism $S_{security\_context}$ = 256 bytes for both RSA and DSA.

Figure 6 shows how the storage cost varies with respect to the number of active mobile producers. For a mobile EPC network the number of mobile users is in the order of 1 million. If we consider the worst-case scenario for the storage cost, i.e., every router has an entry per each user in its forwarding structure, we can see that the storage cost is about 50MB at each router. Modern router device can easily store such amount of data.

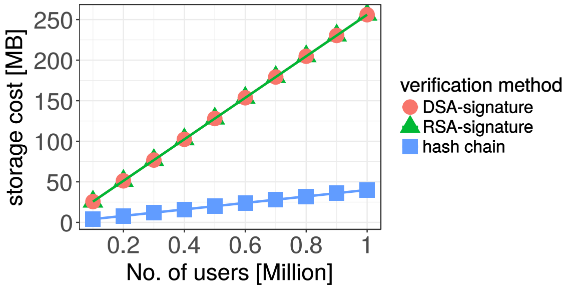


Figure 6: storage cost at each router

To evaluate the storage cost for **MP**, we consider the proposal by Coppersmith and al. [13]. This mechanism requires to store $log_2 n$ number of hashes, where n is the length of the hash-chain. Therefore, if we assume n equal to 1 billion and the size of a single hash of 32 bytes, we only require less than 1KB to store the chain. While this requires more space than storing the single public key for the signature verification approach, we argue that 1KB is a negligible space overhead for most of the nowadays available devices (including IoT devices).

7 CONCLUSION AND FUTURE WORK

The name-based communication model in ICN offers a native support for handling mobility at network layer. Previous work in ICN literature focused on name-based mobility management solutions and particularly on the challenges of producer mobility, which involves an interaction between forwarding and control plane.

In this paper, we took a look at the security implications of producer mobility tracing-based protocols. We present a protocol for prefix attestation based on hash-chain that protects against prefix hijacking attacks occurring during mobility updates. Our protocol is lightweight, fully distributed and it can run unchanged on different hardware deployed at network access (e.g., LTE or WiFi).

Initial evaluation results show that our hash-based approach outperforms the signature based approach, the leading alternative to thwart prefix hijacking attacks. In particular, our approach maintains more than 90% of the original goodput (i.e., without any security mechanism in place), while the signature approach drops it close to 0%. In terms of storage requirements, our protocol only requires tens of megabyte on each router to manage billions of mobile producers.

As future work, we plan to design and implement a secure mechanism for preventing Denial of Service attacks that exploits our protocol as attack vector.

REFERENCES

[1] 2016. OCTEON III CN7020. http://www.cavium.com/new/Table.html#Octeonplus. (2016).
[2] 2017. openwrt benchmark result. https://wiki.openwrt.org/doc/howto/benchmark.openssl. (2017).
[3] Mohammad Al-Shurman, Seong-Moo Yoo, and Seungjin Park. 2004. Black hole attack in mobile ad hoc networks. In *Southeast Regional Conference*. ACM, 96–97.

[4] Moreno Ambrosin, Alberto Compagno, Mauro Conti, Cesar Ghali, and Gene Tsudik. 2016. Security and Privacy Analysis of NSF Future Internet Architectures. *arXiv preprint arXiv:1610.00355* (2016).

[5] Jordan Augé, Giovanna Carofiglio, Giulio Grassi, Luca Muscariello, Giovanni Pau, and Xuan Zeng. 2016. MAP-Me: Managing Anchor-less Producer Mobility in Information-Centric Networks. *arXiv preprint arXiv:1611.06785* (2016).

[6] Aytac Azgin, Ravishankar Ravindran, and Guoqiang Wang. 2014. A scalable mobility-centric architecture for named data networking. *arXiv preprint arXiv:1406.7049* (2014).

[7] Hitesh Ballani, Paul Francis, and Xinyang Zhang. 2007. A Study of Prefix Hijacking and Interception in the Internet. *ACM SIGCOMM Computer Communication Review* 37, 4 (2007), 265–276.

[8] Daksha Bhasker. 2013. 4G LTE security for mobile network operators. *Cyber Security and Information Systems* 1, 4 (2013), 20–29.

[9] K. Butler, T. R. Farley, P. McDaniel, and J. Rexford. 2010. A Survey of BGP Security Issues and Solutions. *Proceedings of the IEEE* 98, 1 (Jan 2010), 100–122.

[10] Andrew T Campbell, Javier Gomez, Sanghyo Kim, András Gergely Valkó, Chieh-Yih Wan, and Zoltán R Turányi. 2000. Design, implementation, and evaluation of cellular IP. *IEEE Personal Communications* 7, 4 (2000), 42–49.

[11] Jeffrey Cichonski, Joshua M Franklin, and Michael Bartock. 2016. Guide to LTE Security. *DRAFT NIST Special Publication 800-187* (2016).

[12] Mauro Conti, Paolo Gasti, and Marco Teoli. 2013. A lightweight mechanism for detection of cache pollution attacks in named data networking. *Computer Networks* 57, 16 (2013), 3178–3191.

[13] Don Coppersmith and Markus Jakobsson. 2002. Almost optimal hash sequence traversal. In *International Conference on Financial Cryptography*. Springer, 102–119.

[14] Subir Das, Archan Misra, and Prathima Agrawal. 2000. TeleMIP: telecommunications-enhanced mobile IP architecture for fast intradomain mobility. *IEEE Personal Communications* 7, 4 (2000), 50–58.

[15] Cesar Ghali, Marc A. Schlosberg, Gene Tsudik, and Christopher A. Wood. 2015. Interest-Based Access Control for Content Centric Networks. In *Information-Centric Networking (ICN)*. ACM, 147–156.

[16] Cesar Ghali, Gene Tsudik, and Ersin Uzun. 2014. Needle in a haystack: Mitigating content poisoning in named-data networking. In *NDSS Workshop on Security of Emerging Networking Technologies (SENT)*.

[17] Dookyoon Han, Munyoung Lee, Kideok Cho, T. Kwon, and Y. Choi. 2014. Publisher mobility support in content centric networks. In *International Conference on Information Networking (ICOIN)*. IEEE, 214–219.

[18] Frederik Hermans, Edith Ngai, and Per Gunningberg. 2012. Global source mobility in the content-centric networking architecture. In *Emerging Name-Oriented Mobile Networking Design-Architecture, Algorithms, and Applications*. ACM, 13–18.

[19] Geoff Huston, Mattia Rossi, and Grenville Armitage. 2011. Securing BGP - A literature survey. *IEEE Communications Surveys & Tutorials* 13, 2 (2011), 199–222.

[20] Stephen Kent, Charles Lynn, and Karen Seo. 2000. Secure border gateway protocol (S-BGP). *IEEE Journal on Selected areas in Communications* 18, 4 (2000), 582–592.

[21] Do-hyung Kim, Jong-hwan Kim, Yu-sung Kim, Hyun-soo Yoon, and Ikjun Yeom. 2012. Mobility support in content centric networks. In *ACM SIGCOMM workshop on Information-Centric Networking (ICN)*. ACM, 13–18.

[22] Leslie Lamport. 1981. Password authentication with insecure communication. *Commun. ACM* 24, 11 (1981), 770–772.

[23] Jihoon Lee, Sungrae Cho, and Daeyoub Kim. 2012. Device mobility management in content-centric networking. *IEEE Communications Magazine* 50, 12 (2012), 28–34.

[24] Dawei Li and Mooi Choo Cuah. 2013. SCOM: A scalable content centric network architecture with mobility support. In *Mobile Ad-hoc and Sensor Networks (MSN)*. IEEE, 25–32.

[25] Sandra Murphy and Madelyn Badger. 1997. *OSPF with digital signatures. RFC 2154*. Technical Report.

[26] Moni Naor and Kobbi Nissim. 2000. Certificate revocation and certificate update. *IEEE Journal on selected areas in communications* 18, 4 (2000), 561–570.

[27] Bart Preneel. 1994. Cryptographic hash functions. *Transactions on Emerging Telecommunications Technologies* 5, 4 (1994), 431–448.

[28] Paul C van Oorschot, Tao Wan, and Evangelos Kranakis. 2007. On interdomain routing security and pretty secure BGP (psBGP). *ACM Transactions on Information and System Security* 10, 3 (2007), 11.

[29] Tao Wan, Evangelos Kranakis, and Paul C van Oorschot. 2004. S-rip: A secure distance vector routing protocol. In *Applied Cryptography and Network Security (ACNS)*. Springer, 103–119.

[30] Liang Wang, Otto Waltari, and Jussi Kangasharju. 2013. Mobiccn: Mobility support with greedy routing in content-centric networks. In *Global Communications Conference (GLOBECOM)*. IEEE, 2069–2075.

[31] Russ White. 2003. Securing BGP through secure origin BGP (soBGP). *Business Communications Review* 33, 5 (2003), 47–53.

[32] Lixia Zhang, Alexander Afanasyev, Jeffrey Burke, Van Jacobson, Patrick Crowley, Christos Papadopoulos, Lan Wang, Beichuan Zhang, et al. 2014. Named data networking. *ACM SIGCOMM Computer Communication Review* 44, 3 (2014), 66–73.

[33] Yu Zhang, Alexander Afanasyev, Jeff Burke, and Lixia Zhang. 2016. A survey of mobility support in named data networking. In *Infocom Workshop on Computer Communications*. IEEE, 83–88.

[34] Yu Zhang, Hongli Zhang, and Lixia Zhang. 2014. Kite: A mobility support scheme for ndn. In *Information-Centric Networking (ICN)*. ACM, 179–180.