IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received July 15, 2020, accepted July 20, 2020, date of publication August 7, 2020, date of current version August 19, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3014898

Public Key Certificate Privacy in VoNDN: Voice Over Named Data Networks

ERTUGRUL DOGRULUK, (Student Member, IEEE), ÓSCAR GAMA, ANTÓNIO D. COSTA, (Member, IEEE), AND JOAQUIM MACEDO
Centro Algoritmi, Department of Informatics, University of Minho, 4710-057 Braga, Portugal

Corresponding author: Ertugrul Dogruluk (d7474@di.uminho.pt)

This work was supported by the Fundação para a Ciência e Tecnologia (FCT) within the Research and Development Units Project Scope under Grant UIDB/00319/2020.

ABSTRACT Named Data Network (NDN) is a network paradigm that attempts to answer today's needs for distribution. One of the NDN key features is in-network caching to increase content distribution and network efficiency. However, this feature may increase the privacy concerns, as the adversary may identify the call history, and the callee/caller location through side-channel timing responses from the cache of trusted Voice over NDN (VoNDN) application routers. The side-channel timing attack can be mitigated by countermeasures, such as additional unpredictable delay, random caching, group signatures, and no-caching configurations. However, the content distribution may be affected by pre-configured countermeasures, which may be against the original purpose of NDN. In this work, the detection and defense (DaD) approach is proposed to mitigate the attack efficiently and effectively. With the DaD usage, an attack can be detected by a multi-level detection mechanism, in order to apply the countermeasures against the adversarial faces. Also, the detections can be used to determine the severity of the attack. In order to detect the behavior of an adversary, a brute-force timing attack was implemented and simulated of the VoNDN application on NDN-testbed. A trusted application that mimics the VoNDN and identifies the cached certificate on a worldwide NDN-testbed. In simulation primary results showed that the multi-level detection based on DaD mitigated the attack about 39.1% in best-route, and 36.5% in multicast communications. Additionally, the results showed that DaD preserves privacy without compromising the efficiency benefits of in-network caching in the VoNDN application.

INDEX TERMS NDN, VoNDN, certificate privacy, a side-channel timing attack.

I. INTRODUCTION

Nowadays, the Internet is forced to handle content distribution for social media and online content services. However, the IP network is not suitable for such use, because of its point-to-point design. To overcome this problem, Information-Centric Networks (ICNs) have been proposed, such as the Named Data Network (NDN) [1], [2]. In NDN, the contents are named and each content is cached by the network nodes to improve the content distribution performance. Nevertheless, caching, despite its benefits, may threaten the privacy of NDN consumers and producers [3]. For instance, content previously cached may reveal its name, content, and signature through a side-channel timing attack [4]. To mitigate the timing attack, diverse countermeasure methods have been proposed by other works, such as artificial content delay [5], [6], random caching [4], and encrypted names [7]. Although these methods are effective to mitigate the side-channel attacks, they may affect the performance of the content distribution.

This work attempts to answer the following question: *Is it possible to maintain the content distribution while using countermeasures to mitigate the side-channel timing attack?* In this work, a trusted Voice over NDN (VoNDN) application was developed to point out the privacy of the certificate. In order to achieve realistic results, the trusted VoNDN application was run on a simulator that emulates the NDN-testbed [8]. An alternative countermeasure approach is proposed (detection and defense (DaD)) to maximize content distribution. The DaD is based on detecting the attack first then apply multi-level countermeasure methods on the attacked router. The scenario results showed that the DaD can be stated

The associate editor coordinating the review of this manuscript and approving it for publication was Giacomo Verticale.

as an efficient method to mitigate the attack and preserving the legitimate nodes compared to other static countermeasure configurations.

The rest of the paper is organized as follows. Section II summarizes the previous related work for the side-channel timing attack in NDN. Section III summarizes the NDN architecture and its data-centric features. Section IV describes the voice-over NDN architecture with its trusted scheme for the consumers. Section V presents the current countermeasure methods. Section VI introduces the Detection and Defense privacy approach to efficiently mitigate the side-channel timing attack on the VoNDN certificate scheme. Section VII presents the attack scenario implementation and discusses the results. Finally, Section IX draws the conclusions.

II. RELATED WORKS

The related works include attacks and countermeasures.

A. ATTACK RELATED WORKS

The works [1], [9]–[11], [3], [7], [12], [13], and [4] presented and discussed possible side-channel timing attack design that may affect the content, signature (certificate), and name privacy in ICN and NDN networks. This design (traditional) is based on particular targets to succeed in the attack. However, such a design may reduce attack performance and efficiency considering multiple targets.

The work [13] presented an attack-type that is for Geo-locating the consumers in the NDN-testbed. The consumer may have the information about hop count which used to obtain the hops between the routers. The adversary may use this information to obtain consumers' cached contents by NDN-testbed hops. This attack is similar to a side-channel timing attack because the cached contents hop counts can be slightly noticeable for non-cached contents. To mitigate the attack, it was prosed that the hop count information may be turned-off for the users.

B. COUNTERMEASURE RELATED WORKS

The work [4] widely studied the cache privacy and the adversary threats to the consumer and producer privacy on the NDN paradigm. Also, countermeasure methods were proposed to mitigate the attack based on the other work [5] -*k* anonymity based delay algorithms (no cache, delay, and random cache). These countermeasure methods applied to privacy-sensitive contents can be indicated by its producer and consumer.

The delay may be used to mitigate the side-channel attack on privacy-sensitive indicated contents. However, an additional delay may imply a trade-off between privacy and latency because it also applied to legitimate requests. For instance, a higher τ value can disable the cache on routers as discussed by [4] and [6]. Also, user-driven countermeasures may not be usable in the real world. For instance, what is private for a user may not be private by other users.

The work [10] presented an extensive study for timing attacks on ICN privacy. The side-channel timing attack and its findings were presented with possible countermeasure methods to mitigate the attack in ICN networks. The trade-offs of countermeasures were evaluated by primary results, especially on additional delay approach algorithms. Also, they proposed a user-driven countermeasure method called "*Vanilla*". For privacy-sensitive contents, an edge router caches the content from the producer and keeps the retrieval times of the first interest and delay the next coming requests. However, the per-client solution will not be feasible, because of a large number of consumers and content distribution efficiency.

The NDN promising maximum in-network caching feature to achieve lower latency for requested contents. However, the work [14] stated that the cache may not be necessary to be configured for maximum size. For instance, the cache can have the same performance on different distributions. Through not caching each content, privacy can be preserved for a timing attack. This method can reduce the attack performance but considering at least one segment must be cached by a router, this can be still targeted by an adversary.

The work [7] presented an anonymity tool called ANDaNA that is build top of NDN. The tool provides maximized consumer anonymity through unreadable (encrypted) name-spaces. However, when the name-spaces became unreadable except whoever asks for it, the usage of CS becomes useless because no consumer can retrieve contents from CS. Similarly like ANDaNA, the work [15] also presented a system tool called PrivICN that relies on encryption schemes on contents. The other work Kondo *et al.* [16] and Chaabane *et al.* [6] proposed a name filtering against information privacy leakages in NDN. Also, the bloom filters may introduce false positives and periodically require resetting and reducing the performance of the cache. Effectively, the un-readable name and filtering techniques may prove "*perfect privacy*" but also comes with the disabled cache. Since NDN promising the contents must be cached to achieve low latency, the name filtering and encryption approaches may not be the most feasible for the NDN paradigm.

The group signatures can be used to preserve the signature (certificate) privacy as proposed by Chaum and Van Heyst [17] and Boneh *et al.* [18]. A group signature can be done by any member of a group. The signature is verifiable by anyone, but it is not possible to know which group member actually done it. Consequently, the privacy of the certificate or content can be maintained in the trusted group. However, considering collaborating with the group members, such a group trust scheme relies on limited configurations for verification and identification.

III. NAMED DATA NETWORKING

The NDN architecture aims to develop a new Internet architecture that can capitalize on the strengths and address weaknesses of point-to-point architecture to naturally accommodate emerging patterns of communication needs.

The works of [1], [19], and [2] have proposed the NDN project, a network paradigm that is an evolution of the IP architecture. In NDN, any packet object can be named instead of naming the endpoints. This is a feature that changes the network semantics from packet delivery to the identified destination address to caching data by a given name.

NDN is based on two main packet types: *interest* packets and *data*[1] packets. Interest packets are issued by the consumers and data packets by the producers. The content name in the interest packet identifies the request of the consumer, for example, */pt/uminho/algoritmi* is a name for a content, expressed in a structured way. It can be used to name contents related to the Algoritmi research center. As shown in Figure 1, the producer includes a signature in the data packet. Mechanisms for signing and verifying the integrity of the contents have been proposed for NDN, such as the one described in [1], [19].

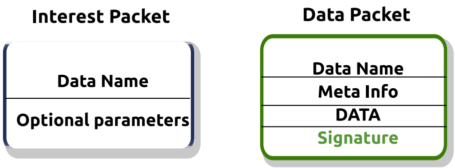


FIGURE 1. Packet types in NDN (adapted from [2]).

A. IN-NETWORK STORAGE

NDN supports that any data packet can independently be retrieved from the network. Thus, the NDN router can cache data packets in CS (Content Store), to satisfy future incoming interests for contents. The CS is similar to today's buffer memories in IP routers. However, each NDN router can reuse a data packet while the IP routers cannot. Note that NDN handles the repositories (*e.g.* CS) and network channels as data retrieval sources.

On the other hand, the CS is a benefit for network congestion. In the presence of congestion, if it occurs for any reason, CS retransmits the data. Imagine two congested links along a path between a consumer and a producer. If the requested data packet gets through the first congested link somehow, but couldn't get through the second one, then the data packet is dropped. But it remains in the CS of the intermediary node. Then, the consumer's interest becomes timed-out and the interest packet is resent. Caching will allow the data packet to be retransmitted to the consumer over only the second congested link. However, on the traditional Internet, the retransmission of the data packet can only be done by the content producer, and the data packet has to pass through the first congested link again.

So, CS presents optimal data delivery for static content, while getting supported by today's in-network repositories without having an application layer overlay. Even the dynamic contents, such as broadcasting or real-time conferencing, can benefit from CS in case of a packet loss.

B. ROUTING AND FORWARDING

In an NDN router, the forwarding of *interest* packets and *data* packets are carried out by three engines: Pending Interest Table (PIT), Forwarding Information Base (FIB), and Content Store (CS).

CS represents a cache for data packets, similar to Internet routers buffer memory. FIB is a name prefixes routing table and respective outgoing interfaces, used to route interests. PIT is a pending interest table and a set of corresponding incoming interfaces.

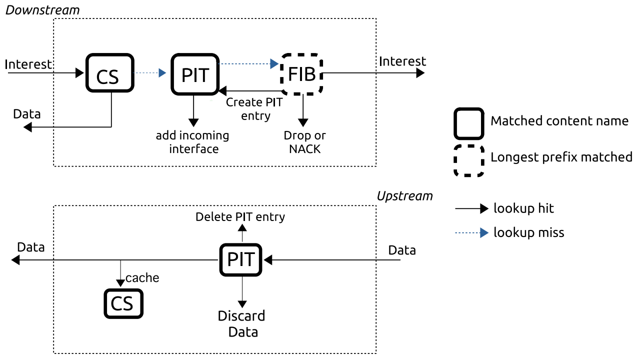


FIGURE 2. NDN forwarding engine model (adapted from [1]).

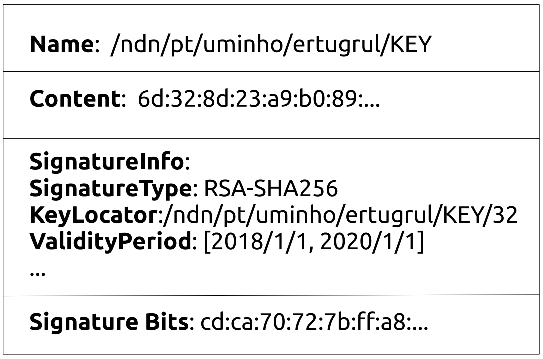
As Figure 2 illustrates, when a consumer interest is received, a lookup is made on the CS for previously cached content that matches the name requested. If there is a matching content, it is sent as a reply with a data packet. If the data packet is not in the CS, the router checks the data name in the PIT. If there is a matching name in the PIT, the router records the incoming face of the interest for the future reply and stops the procedure. If not, a new PIT entry is created for that name, recording the incoming interface, and a lookup is done on FIB. The interest is routed using FIB information to the producer. For each forwarding interest packet, the longest name prefix match is a lookup in the FIB, which determines where to send it. The list of outgoing faces of the FIB matched entry is an important reference for the routing. In case a name is not found in FIB, the interest becomes unsatisfied. When all FIB lookup misses are replied with a NACK packet, the forwarder can limit the requests [1], [19]. For example, this can be used to mitigate the denial of service attack in NDN, as described in [20].

C. SECURITY AND PRIVACY

In NDN, security is a built-in data function, rather than being a function of how and where data is obtained. The producer signs the data packet binding its name to data. In NDN, it is mandatory to use the signatures. So, each data packet and corresponding application must meet security requirements.

NDN supports a fine-grained trust scheme. This allows determining whether a public key owner is an acceptable content publisher, for a specific piece of the data, into a particular context.

[1] In this work, the terms "*data*" and "*content*" are used interchangeably.



Name: /ndn/pt/uminho/ertugrul/KEY

Content: 6d:32:8d:23:a9:b0:89:...

SignatureInfo:
SignatureType: RSA-SHA256
KeyLocator:/ndn/pt/uminho/ertugrul/KEY/32
ValidityPeriod: [2018/1/1, 2020/1/1]
...

Signature Bits: cd:ca:70:72:7b:ff:a8:...

FIGURE 3. NDN certificate format (adapted from [21]).

Figure 3 illustrates the NDN certificate. In NDN, a certificate can be presented as like any other content that is carrying a public key. The "key" refers to a content packet that carries a public key. The *KeyLocator* refers to the certificate issuer or certificate authority.

A data packet seals the binding between name and data through a digital signature. The advantage of using the certificate as a data packet is that a consumer can retrieve and validate a certificate by issuing an interest packet. Indeed, the public key certificate has the general format of a data packet. For instance, a producer expresses a certificate challenge using name and content to carry the public key bits.

Content-centric security and trust ensure the integrity of the content. However, the in-network storage feature increases privacy concerns. Also, the digitally signed packets do not guarantee to protect content or cache privacy in NDN. For instance, the recently cached content characteristics may be different than un-cached ones, the adversary uses this information to determine the cached contents from CS. The information may be present publicly the public key certificate, the name, the content, and the cache size.

All NDN contents are digitally signed by its producer, to provide integrity and guarantee on provenance which makes all signatures publicly verifiable by the NDN nodes and application layers.[2]

However, digitally signed contents are may leak sensitive information about the content signer. Because contents carry a public key that is publicly fetchable by any consumer, the adversary may be able to determine content producer by using a timing attack.

For instance, the two-way conversation tool VoCCN [22] is based on signed contents, to keep content integrity between the callee and caller.[3] In this structure, the certificates may be used to verify the content producer. Since the certificates are effectively a content and cached in NDN node by the application layer, the adversary may use the timing differences, to identify the call history, location, and users by a timing attack.

IV. VOICE OVER NDN

Voice over IP (VoIP) is a transmission method of voice/video communication over IP. VoIP requires intermediary exchange protocols, such as Real-Time Transport Protocol (RTP) or Secure RTP (SRTP), and a signaling protocol, *e.g.* Session Initiation Protocol (SIP) to establish a call, as described in [23].

In CCN networks, the data flows directly from the producer to the consumer. Therefore, the media and signaling paths can be defined between the producer and the consumer. Based on this idea, the VoCCN design has been proposed by [22]. In VoCCN, the signaling and media paths can be combined and the voice packets can directly flow between callee and caller without requiring any translation middleware because packets can flow directly between callee and caller.

Architecture of Voice Over NDN (VoNDN): Similarly to VoCCN, this work introduces voice-over NDN (VoNDN) as a use case for testing purposes.

As Figure 4 illustrates, the NDN name can be used to establish signaling and media paths for voice/video calls. The SIP may be used to create a signal path from Alice to Bob. A SIP invitation message carries a randomly generated symmetric key *k*. The caller (Alice) can encrypt the key block *(k)* using callee's (Bob) public key (B_pub). When creating a signaling interest packet the caller would include both the encrypted block (B_pub(*(k)*)) and the authenticated SIP message (*(k)*(SIP_INVITE)). The callee, on receiving the interest, could decrypt the key block with its private key, recover *(k)*, and use it to verify and decrypt the SIP_INVITE. The caller would then use key *(k)* to encrypt its SIP_RESPONSE message.[4]

When the signaling path is securely established, the SIP packets are replaced by RTP media packets. As seen in Figure 4, the SIP exchange section is replaced by the call identification (call-id), together with other required information and a sequence number (seq-no) used to control different media fragments.

A. POTENTIAL PRIVACY RISKS IN VoNDN

Theoretically, the secured VoNDN conversation may face side-channel timing attacks. The aforementioned attributes of encrypted traffic (trusted conversation) of the side-channel information may be used to leak insights from the communication users. Since the callee and caller are presented as producer and consumer, the cached contents may be used to identify the callee or the caller (conversation pairs), the location, and the time of the established conversation. For instance, Zhang *et al.* [24] show how to reveal the voice call history of the user by side-channel timing attack in IP. The attack method also can reveal the call history of a group call in VoIP. The timing attack is aimed against the victim's SIP proxy server. The other work Lauinger [9] and Lauinger *et al.* [12] studied the side-channels on Voice-over CCNs. It is shown

[2] Currently the NDN offers RSA (Rivest–Shamir–Adleman) and ECDSA (Elliptic Curve Digital Signature Algorithm) signature algorithms.

[3] The callee and caller is an effective content producer and consumer.

[4] In VoNDN, the *callee* and the *caller* play the role of producer and consumer at the same time.

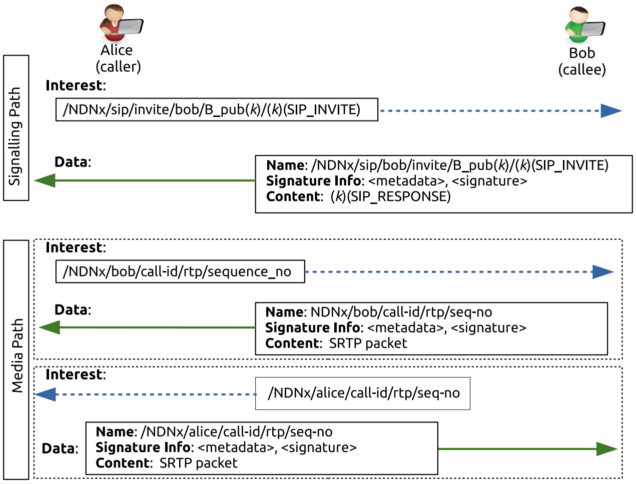

Alice
(caller)
Bob
(callee)
Signalling Path
Interest:
/NDNx/sip/invite/bob/B_pub(k)/(k)(SIP_INVITE)
Data:
Name: /NDNx/sip/bob/invite/B_pub(k)/(k)(SIP_INVITE)
Signature Info: <metadata>, <signature>
Content: (k)(SIP_RESPONSE)
Media Path
Interest:
/NDNx/bob/call-id/rtp/sequence_no
Data:
Name: NDNx/bob/call-id/rtp/seq-no
Signature Info: <metadata>, <signature>
Content: SRTP packet
Interest:
/NDNx/alice/call-id/rtp/seq-no
Data:
Name: /NDNx/alice/call-id/rtp/seq-no
Signature Info: <metadata>, <signature>
Content: SRTP packet

FIGURE 4. **VoNDN combined paths (adapted from [22]).**

that an adversary can replicate the VoCCN packets in order to learn the size of the cached voice packets. Because of the voice is encoded using by variable-bit-rate encoding scheme, each of voice packets can be shaped by its phrases. This may lead to learning the previously spoken voice packets even the conversation is encrypted between VoIP pairs as described by [25], [26].

In VoNDN, the voice packet public-key integrity is established by the certificate and may be managed by a SIP authentication domain, such as an inter-domain authentication protocol (*e.g.* IP authentication protocol RFC4474 [27]). To expedite the next request(s), the certificate is cached for a certain period by a SIP proxy.

As Figure 5 illustrates, the certificate from the caller's domain is cached by callee's proxy. The adversary takes advantage of the SIP processing time to obtain a certificate that has been cached or not. Through the time responses of certificates, the adversary may obtain the VoNDN call history of a SIP domain.

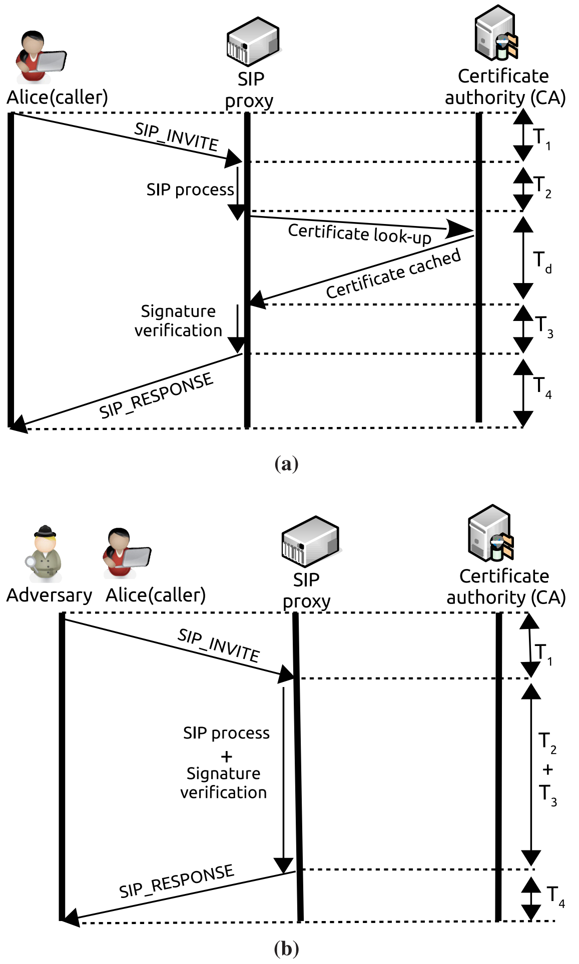
In this work, the Content Retrieval Time (CRT) is defined as the period between sending the interest and retrieving the respective content, which can be cached or un-cached. As certificates are treated as contents, the CRT definition can be also applied for the certificates.

Figure 5a illustrates that the caller has to get the certificate from the certification authority since it is not cached by CS. This certificate request process time is calculated by the equation:

$$CRT_{uncached} = T_1 + T_2 + T_d + T_3 + T_4 \tag{1}$$

where:

T_1: the content retrieval time for sending the SIP_INVITE message.
T_2: the processing time for the SIP message.
T_3: the signature verification time for Alice's identity.
T_4: the CRT time response from Alice SIP proxy (*e.g.* content store).
T_d: the CRT time for the intended lookup certificate.



(a)

(b)

FIGURE 5. **Side-channel timing attack on VoNDN trust scheme, (a) first time for certificate lookup, (b) Future request the certificate from CS (adapted from [24]).**

As illustrated in Figure 5b, if Alice's certificate has been cached by the CS, then its request process time is calculated by the equation:

$$CRT_{cached} = T_1 + T_2 + T_3 + T_4 \tag{2}$$

In this example, the adversary targeted Alice's certificate by distinguishing the cached and un-cached certificate CRT responses.

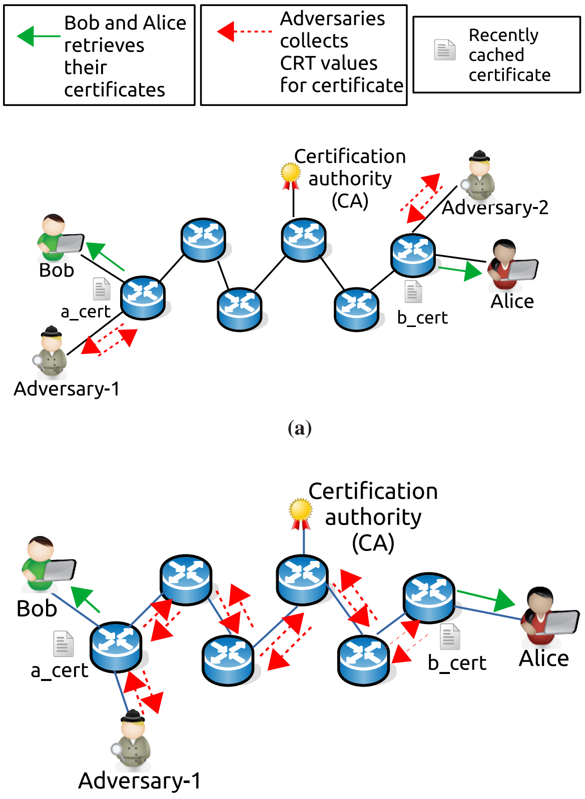
Note that, only the caller's (Alice) side is illustrated in Figure 5. On the other side, similarly, the callee (Bob) receives the invitation from the caller that needs to be verified and fetch with the caller's (Alice) certificate from the SIP server. Thus callee can validate the signature of the caller to establish the call.

B. DETERMINE CLOSE AND AWAY TARGETS

In naive condition, the certificate can be replied by corresponding CS to the callee and caller in VoNDN. The adversary may use cached certificate CRT value to determine the consumer location or established call time. Note that, if the certificated packet only issued for a certain consumer, the adversary cannot determine the content because of a lack of the private key. Still, the adversary can knowledge the existence of the location of the cached content because of knowing the public key.

On the other hand, the adversary can determine the distance of a certificate from its cache using the CRT information. Let us suppose that the side-channel timing attack CRT measurement for a certificate is CRT_2 (retrieved content from CA), CRT_1 is the CRT from the edge NDN router, CRT_e is the expected CRT of the intended content lookup, and ε is a very small time difference. According to Chaabane *et al.* [6] and Mohaisen *et al.* [10], after collecting the CRT samples, the adversary concludes that:

- if $|CRT_\text{e} - CRT_1| < \varepsilon$, the target certificate has been cached by the edge router.
- if $|CRT_\text{e} - CRT_2| < \varepsilon$, the target certificate is not cached by any router, except the certificate authority.
- if $CRT_\text{e} > CRT_1$ and $CRT_\text{e} < CRT_2$, the target certificate has been cached by away routers. Note that, the adversary can still predict the certificate location (number of hops) by relying on CRT_1 and CRT_2 values.


Bob and Alice retrieves their certificates
Adversaries collects CRT values for certificate
Recently cached certificate
Certification authority (CA)
Adversary-2
Bob
Alice
a_cert
b_cert
Adversary-1
Certification authority (CA)
Bob
a_cert
b_cert
Alice
Adversary-1

(a)

(b)

FIGURE 6. Side-channel timing attack on close and away targets: (a) Determine Bob's and Alice's certificate from the edge router location, (b) Determine Bob's certificate from away router location.

The CRT side-channel timing attack has been based on two adversary models: *i*. one able to identify the certificates at the edge (closest) router (Figure 6a), and *ii*. another able to identify the cached certificate from away (more than one hop) routers (Figure 6b). When an adversary collects all possible CRT values, it can do three assumptions about where the certificate (target) has been cached.

First, the maximum CRT value shows that the certificate is not cached by any NDN routers expect CA. Second, the minimum CRT value indicates the certificate has been recently cached by the edge router. Third, if the CRT is between a minimum and a maximum value, the adversary concludes that the content was cached by an away router. Note that, this router can be one hop (neighbor) or more than one hop away.

Additionally, more than one adversary can be used to determine the established call between Bob and Alice by attacking their certificates (Figure 6a). On the other hand, a single adversary may still determine Bob's certificate from away location (Figure 6b).

According to the cached certificate CRT comparisons, the scope of the trusted VoNDN attack can be summarized by following determinations:

- Figure 6a illustrates that the adversaries (Adversary-1,-2) estimate the time of an established call between Alice and Bob from their edge routers by looking up their certificates (a_cert and b_cert).
- Figure 6b illustrates that the adversary (Adversary-1) identifies that who had a conversation with Bob recently by looking up to Alice's certificate (a_cert) from edge router. Also, the approximated the location of Alice by looking up to Bob's certificate (b_cert) from away location.
- Figure 6a and 6b are illustrated that the adversaries (Adversary-1,-2) identify where the call was established between Bob and Alice by comparing CRT responses of the certificates (a_cert and b_cert).

V. COUNTERMEASURES

In VoNDN, the side-channel timing attack can be mitigated by statically (always-on) pre-configured countermeasures on the caching routers. These can be based on the manipulate of the CRT values of the NDN router. Through this, the adversary may not able to obtain the shortest CRT values which also used to obtain the cached certificates. In this work, the countermeasure methods were classified by *i*. cache available and *ii*. cache disabled approaches.

Figure 7 illustrates the main concept of the countermeasure approaches. In this scenario, Alice requested content from the producer and it replied the content segments, were cached by the router and CRT calculated as Δ+t. If Alice re-request a segment, the cache replies to this request instead of sending it till the producer and CRT calculated as Δ. In the side-channel timing attack, the adversary pursues the CRT of Δ to obtain the target that has been cached recently.

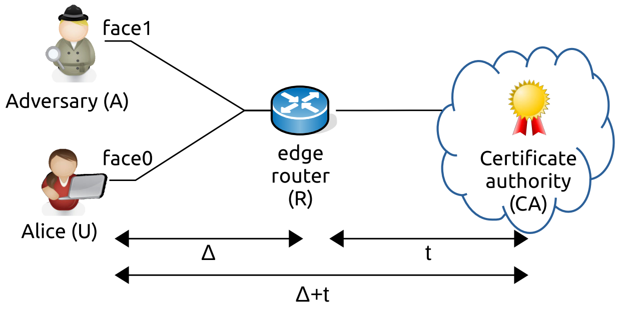


FIGURE 7. Statically configured countermeasure configurations.

Since the Δ can be used to illustrate for all cached contents and the adversary pursues it, the countermeasure methods can be based on increasing the CRT value of Δ with some additional value from the configuration, as discussed next. Based on the countermeasure configurations, the attack can be mitigated on adversary face (*face1*). However, these countermeasures based on the static configuration which also affects the legitimate node requests (*face0*). Therefore, the pre-configured countermeasures are not able to distinguish between adversary and legitimate nodes.

A. CACHE AVAILABLE METHODS

The cache available countermeasure methods are used to increase the value of Δ to all faces (face0 and face1) and these can be classified into three groups: *i*. delay content, *ii*. random caching, and *iii*. group signatures.

1) DELAY CONTENT

Data delivery in the NDN is affected by a certain delay imposed by the routers. This delay can be a solution to prevent cached content attacks. Let us consider Δ the default delay value which presents the CRT of cached targets. In a side-channel timing attack, the adversary tries to figure out the Δ value. The adversarial CRT calculation can be challenged if a delay of τ was chosen based on random function by the router. In this case, expected CRT is increased for the adversary, which concludes that the target has not to be cached by the edge router. However, an additional delay to Δ may reduce the content distribution for the cached contents.

On the other hand, because of the attack repetitions, the adversary may find a delay of τ by analyzing the CRT samples from the router. The value of τ can be changed by proposed algorithms. For instance, instead of using a constant τ, Schinzel [28] proposed a τ value based on cryptographic (unpredictable) function. Through, the unpredictable delay function of the τ, the adversary may not able to obtain the cached targets in the router.

A similar delay countermeasure method was proposed by work Acs *et al.* [4] to preserve privacy in NDN. The delay was applied to all faces to mitigate the timing attack in NDN. This delay can be classified by three configurations: fixed, randomized, and unpredictable. To improve distribution efficiency, the delay is configured only for the first requests. Therefore, the first adversary's requests miss the cache and the attack may not succeed for the cached target as proposed by [4] and [6].

2) RANDOMLY CACHING

To reduce the cache redundancy, the capacity of the cache can be defined using probabilistic order as presented by [29]. On the other hand, the probabilistic cache can be used to mitigate the side-channel timing attack. The router can be configured for randomly caching, one named as may be cached another may not be cached depending on the probability configuration. The contents can be also cached by random anonymity set (k), to mitigate the side-channel attack in NDN. Acs *et al.* [4] and Chaabane [6] proposed a random caching, that selects the contents by depending on the random number (k) to mitigate the attacks on the edge router.

Thus, the index of the first cache hit in the output sequence is expected to be random and ideally should not leak information about the router's cache. However, the adversary may also learn the anonymity set value after various cache miss attempts. Therefore, the random value can not be fixed, it should rely on various k-anonymity set (*e.g.* probability rate).

3) GROUP SIGNATURES

In a side-channel timing attack, the adversary uses the CRT estimation to know where the certificate is cached. Cham *et al.* [17] proposed a *group signature* to make public signatures (*e.g.* public key certificate). The signature and the certificate can be associated with a group of users, but not to a specific member of that group. The receiver knows that the signature is valid for any group member. For instance, a conference video/call may use a group signature for privacy protection because the adversary cannot know which member of the group is doing the call.

In other work Boneh *et al.* [18], proposed a short group signature scheme. In this approach, the main goal is to provide the security level of RSA signatures while reducing the length of the signature to accelerate the verification in the group.

In both methods, the certificate can be cached only with a group member to achieve "*perfect privacy*" for the certificate.

B. CACHE DISABLED METHODS

The cache disabled countermeasure approaches offer a "*perfect privacy*" by fully supported anonymity tools. However, the disabled caching approaches can be completely against the NDN paradigm, considering the content distribution must have to be maintained with in-network caching on NDN. In disabled cache approaches, the CRT is obtained as $\Delta + t$ (Figure 7), which is considered as the maximum delay for Alice (face0) and Adversary (face1) face.

1) TURNED-OFF CACHING

In NDN, the CS can be configured for not caching. If there is no content held in the cache, the side-channel timing attack cannot be done. However, the cache is important for the NDN, as it is required for content distribution. So, directly giving up on the caching is not a good option in NDN, as discussed in [4].

2) ANONYMOUS NAMED DATA NETWORKING APPLICATION

The Onion Routing (Tor) was employed layers of concentric encryption and intermediate nodes responsible for peeling off layers as packets travel through the overlay which is commonly referred to as onion routing as proposed by [30]. DiBenetetto *et al.* [7] developed the Anonymous Named Data Networking Application (ANDaNA) a tool to mitigate timing attacks in NDN. ANDaNA is another practice of Tor, built on top of NDN, that provides privacy and anonymity to the consumers. With this tool, the requested names are encrypted and then verified by the nodes and delivered to the user as data. In particular, ANDaNA mitigates timing attacks from linking the retrieved contents in CS. ANDaNA relies on multiple paired-centric layers of encryption and routes content from the consumer via a chain of routers. First, the router decrypts received content then it forwards the content to the next router.

3) PrivICN

The PrivICN is a tool based on name encryption similar to ANDaNA. The tool encrypts the name components except for the longest prefix of the content as presented by [15]. Therefore, the cache is partially available only for the longest prefixes. However, the adversary still can succeed in the attack considering the longest name prefix target to locate target locations.

4) BLOOM FILTERING

The name privacy can be maintained by bloom filters as presented by [6]. In this approach, the consumer can compute hierarchical bloom filter as HB $= (B_1, B_2, \ldots, B_n)$, where B_n is the bloom filter of name component up to n-th component. For example, a consumer can compute a filter B_1 of */ndn*, B_2 of */ndn/pt*, and B_3 of */ndn/pt/minho* for the content of */ndn/pt/minho*. Thus, a router can check the filter B_n from the cache, if it cached it replies to the consumer. If not, the router checks B_n in PIT. If B_n existed in PIT, the bloom filter of the corresponding PIT is updated (add one) and the interest dropped since a request has already been forwarded. Otherwise, it follows the usual NDN paradigm. With this approach, the name in the interest request is obfuscated resulting in transforming it into a random string of bits. However, bloom filters can introduce false positives in name matching.

Kondo *et al.* [16], also presented a similar filter-based approach to preserve name privacy in NDN. This work distinguished the legitimate requests from others based on the filtering. Through the bloom filters, the name of content becomes unreadable because of the human-readable name was transformed in a random-looking string of bits.

VI. DETECTION AND DEFENSE (DaD) PRIVACY MODEL

Effectively, the side-channel timing attack can be mitigated by pre-configured countermeasure configurations, which were previously discussed (available cache and disabled methods). However, these methods may not be the most efficient ones, considering their configuration is static. This may reduce the certificate distribution efficiency for legitimate requests as described by [6], [7], and [10]. Thus, the countermeasures configurations can be considered as a trade-off between privacy and certificate distribution efficiency. To maintain the certificate distribution efficiency and protect the cached certificates this work proposes an approach called detection and defense (DaD).

The DaD is based on attack detection that can distinguish between legitimate and adversary faces. Through this adversary detection, the countermeasure method can be only applied to adversary detected face and legitimate requests can be preserved, without being affected by the available countermeasures. Also, the detection can be used to identify the severity of the attack. Then, different countermeasures can be applied to mitigate the attack. The DaD is based on three attack detection phases where available cache methods are applied in first and second phases and disabled cache in the third phase.

The DaD identifies the attack in three phases as follows: *i. minor phase*, where the attack is detected in the first detection phase period window (TIME) and sets the adversary's face configured with the available cache countermeasure for a time period, *ii. moderate phase*, where the attack persists in the second detection phase and sets the adversary's face configured with a more effective available cache countermeasure compared to the first detection phase, and *iii. severe phase*, where the attack detected in the third detection phase, and the adversary's face configured with the most effective countermeasure (disabled cache) to mitigate the attack.

A. ADVERSARY FACE DETECTION

In this work, side-channel timing attack detection methods were surveyed to detect an attack. However, the number of related work in timing attacks on NDN is limited. For this reason, this work focused on to cache pollution attack detection method which has attack similarities with the side-channel timing attack in NDN [11]. In the cache pollution attack, the adversary may request the same content multiple times to disable the cache function of the router. Also, the adversary may create fake popularity for the contents to interfere with the content distribution performance.

To detect an attack, the works [31] and [32] proposed that the cache hit ratio (CHR) can be used to detect the attack in cache pollution-related attacks. Due to the attack similarities between cache pollution and side-channel timing attack, the DaD attack detection is based on CHR calculations.

In DaD, the detection is face-based. The detection methods can detect the possible adversary face by getting metrics from NDN Forwarding Daemon (NFD). NFD is used as a network forwarder in the network layer. Therefore, the DaD can be used to distinguish between legitimate and adversary nodes to apply multiple countermeasures methods only to the adversary detected face.

1) CACHE HIT RATIO (CHR)

In VoNDN, if a previously requested certificate has been cached by a router, then a cache hit will occur in the next incoming request for the same certificate received by that router. In a side-channel attack, the cache hit ratio (CHR) of the router increases when the attack succeeds [4]. Therefore, the CHR can be used to identify the attack, as proposed by [33]. The CHR can be used to obtain: *i*. the performance of the attack, and *ii*. the identification of the router face under attack.

Ideally, the cache hit ratio can be calculated periodically by an NDN application (*e.g.* NFD) to identify the face of the router that is being attacked in VoNDN. The average CHR of all edge routers is calculated by the total cache hits of each edge router using the following equation:

$$CHR = \frac{\sum_{k=1}^{n} (total\_cache\_hits)_k}{R} * 100\% \tag{3}$$

where n is the total number of the edge routers in the network, and R is the total number of requests received by the edge routers, which is equal to the total number of cache hits plus the total number of cache misses.

The adversary may repeat the request to increase the success of the attack. Thus, the CHR can be used to measure the performance of the attack configuration. Previously, our work studied CHR attack performance calculation for name privacy of the streaming-like NDN application [34]. The results showed that the CHR increased if the adversary succeeds in the attack for previously cached contents.

Besides the attack performance calculation, the CHR can be used not only to detect an attack but also to identify the face of the router that is being attacked. For instance, the CHR can be used to identify the adversary in an attack detection for cache pollution [32] and [31]. Also, the works [11], [12], and [33] proposed that the side-channel timing attack can be identified by cache hits and misses.

In DaD, the attack detection is based on CHR calculation per face for VoNDN application. To detect the adversary's face, the DaD uses the CHR threshold value. If a face's CHR value calculated is higher than the threshold, that face of the router is considered an adversary. Thus, the DaD distinguishes between legitimate and adversary nodes to apply countermeasures.

2) CHR THRESHOLD CALCULATION

In DaD, the CHR threshold parameter is used to identify the face of the router that is being attacked. This parameter is calculated as follows. A set of m requests is collected regularly during ΔT seconds. The total number of cache hits is calculated for the new set of requests, which we consider to be the i^{th} collected set. So, the average CHR of this new set (chr_i) is calculated by the following equation:

$$chr_i = \frac{\sum_{k=1}^{m} CH_k}{m} \tag{4}$$

where CH_k represents the cache hit of the k^{th} request in the new set. The CH_k is one if the k^{th} request gets a cache hit and zero in case of a cache miss. Then, the new global average CHR_j is computed by the following weighted moving average equation:

$$CHR_j = (\alpha \times CHR_{j-1}) + (1 - \alpha) \times (chr_i) \tag{5}$$

where CHR_{j-1} represents the last CHR value, chr_i is the new value calculated by Eq. 4, and α is a weight factor between 0 and 1. The CHR_j is very sensible to the new chr_i value if α is close to 1, or little sensible if α is close to 0. In DaD, α should be chosen close to 0, because an attack increases the CHR_i when it is established, and so the system can detect it quickly. For this reason, α was set close to 0 in our experimental NDN-testbed scenario. The router is considered under attack if CHR_j is higher than the threshold CHR. The CHR thresholds were identified as follows: 5% CHR in VoNDN multicast and 1% CHR in VoNDN (best-route) scenarios. Note that, these thresholds can be either defined manually or dynamically by an algorithm based, for instance, on machine learning techniques.

B. COUNTERMEASURES IMPACT AND SEVERITY OF ATTACK

The face of the router that is being attacked can be identified by the detection methods to apply countermeasures. However, configuring the router with static countermeasures may not be the most appropriate approach, considering that each of the countermeasures configuration effects can be different on mitigation and distribution efficiency.

In DaD, an adversary detected face can be set with different countermeasures. When the adversary detected by CHR, the countermeasure is only applied to the possible identified face. The side-channel adversary detection by CHR can be used: *i*. to configure a countermeasure when the adversary is detected, and *ii*. to determine the severity of the attack which can be used to set different countermeasures.

1) COUNTERMEASURE IMPACT

The DaD is based on three naïve countermeasures to apply available cache configuration by the following:

1) *unpredictable delay*: The attack can be mitigated by uniform (fixed), random distribution, and unpredictable delays. However, the unpredictable delay may be the most challenging to solve by the adversary that is compared to other delay distributions (fixed and random). Thus, the DaD offers an unpredictable delay to mitigate the adversary.
The unpredictable delay is calculated by a hash function [28] and [35]. This function can be expressed by k mod m $=h(k)$, where k is a cryptographic integer hash code from the key that generates unpredictable delay

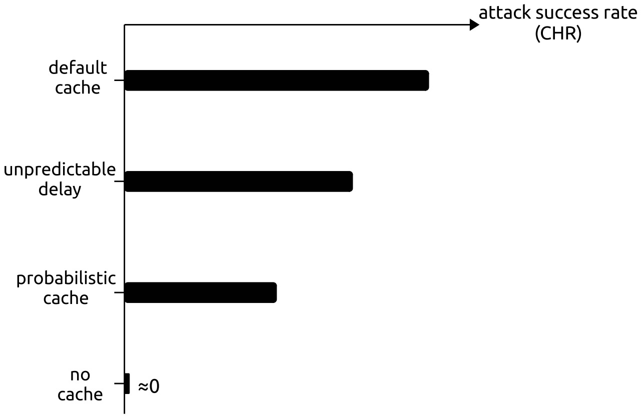


FIGURE 8. A qualitative analysis of the countermeasures impact on the attack success.

h(*k*). Then, this delay is added to the adversary detected face to challenge the adversary CRT calculations.

2) *probabilistically caching*: The edge router can cache the contents by probabilistically [29]. This can be used to mitigate the brute-force attack. The DaD offers the contents that can be cached by p=10% probabilistically selection to the adversary detected face. This means that the edge router is randomly selecting (based on probabilistic function) the content from its cache and answering (random distribution) to the adversary detected face instead of answering directly.
Note that, in case of a detected face is directed to the multiple addresses (devices), legitimate consumers also are affected by the probabilistic cache. In this case, the edge router can probabilistically cache for all faces because it can be a challenge to distinguish between adversary and legitimate faces.

3) *no caching*: This completely disables the cache and the adversary cannot succeed any attack.

Figure 8 compares qualitatively the success of an attack based on default and countermeasures configurations efficiency to mitigate the attack. In this graph, the success of the attack is analyzed based on the cache hit ratio (CHR). For instance, the attack cannot have any success rate in no-cache configuration because it completely disables the cache.

On the other hand, the additional unpredictable delay can be considered to have a better distribution efficiency than the probabilistic cache, because the contents can be already cached by the router in the delay configuration. In this case, the adversary may determine that additional delay by multiple trails of attacks. However, the cache cannot be fully loaded by a randomized distribution configuration. Therefore, in an ideal situation of attack, the unpredictable delay configuration can be considered less effective compared to a randomized distribution configuration.

Note that configuring the router with a high unpredictable delay or low probabilistic rate of caching may affect severely the content distribution efficiency.

2) DETERMINE THE SEVERITY OF THE ATTACK

In DaD, the CHR can be also used to identify the severity of the attack. For instance, if the attack is detected in a period (TIME) and continued in the next detection states, the attack can be considered severe. Note that, the detection period can be tuned by application (*e.g.* VoNDN) configurations. For instance, the adversary was not detected in a higher attack check time because it was already completed the attack. On the other hand, a shorter check time may slow the process of the router. Therefore, through several simulation experiences, an optimum DaD period check attack time was defined as 0.2s for VoNDN applications.

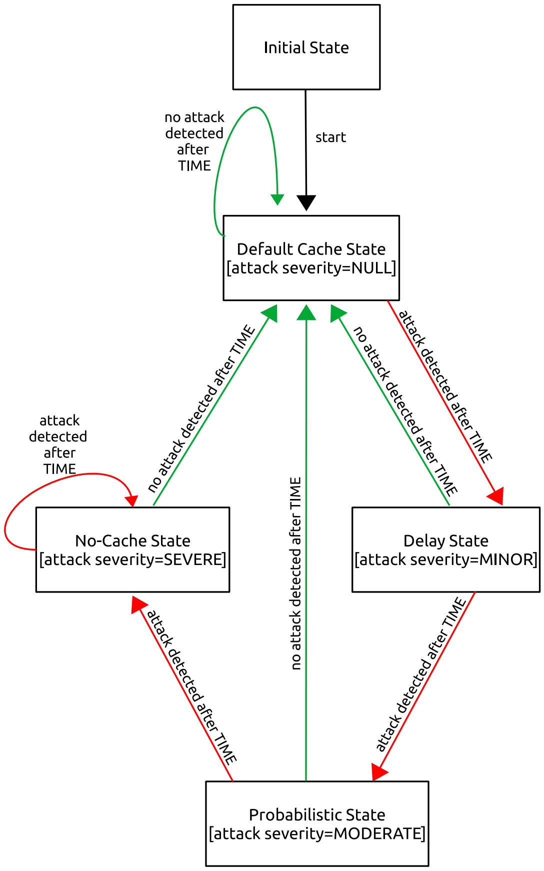


FIGURE 9. Attack states (phases) and applied countermeasures.

According to the attack severity, different countermeasures methods can be applied to mitigate the attack and maintain the content distribution, as illustrated in Figure 9. For instance, the unpredictable delay can be applied in the first detection state (*minor*), the probabilistic caching in the second detection state (*moderate*), and no-cache in the last state (*severe*) to mitigate the effect of the attack. Also, when a detected attack is severe, the no-cache countermeasure is applied while the attack persists. If the adversary withdraws the attack in any detection state, the router is set to the default state (*e.g.* default LRU caching).

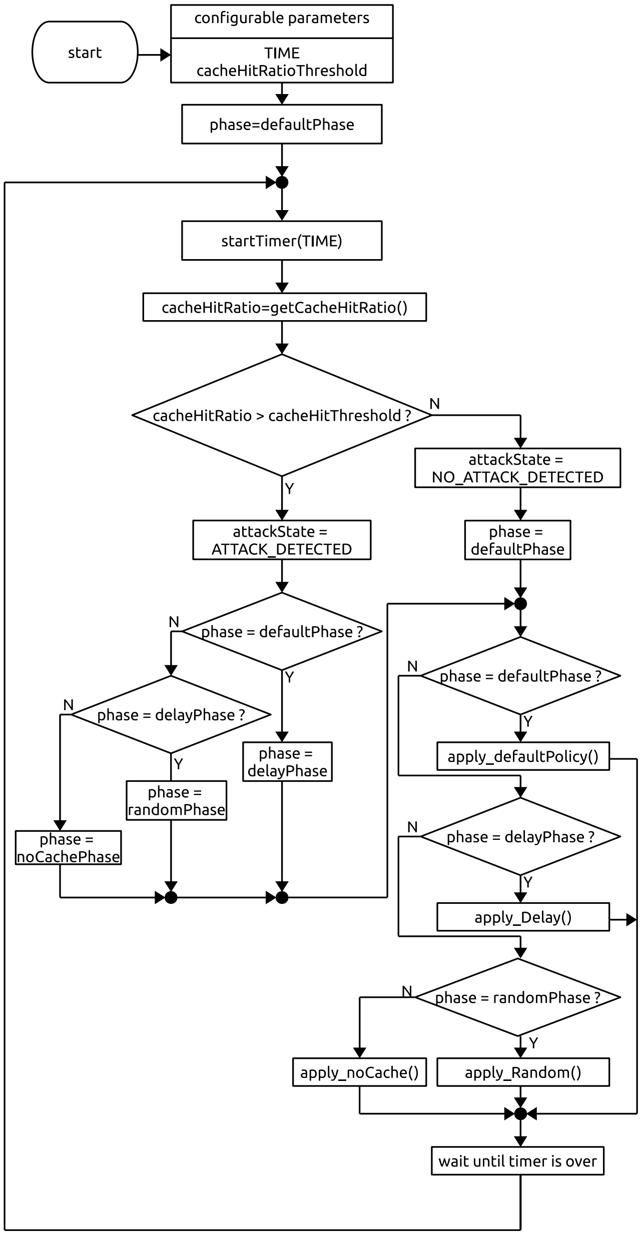


FIGURE 10. Detection and Defense (DaD) flowchart algorithm.

3) DETECTION AND DEFENSE (DAD) ALGORITHM

DaD can be designed for any NDN application to maintain certificate privacy and content distribution. In fact, instead of setting the router with a static countermeasure method, DaD applies dynamically three countermeasures when the router is under attack.

In this work, the DaD algorithm was designed for both name and certificate privacy in a trusted VoNDN and streaming NDNtube applications. Ideally, the algorithm can be implemented in the application layer to protect cached streamed content and certificates. This design is based on the CHR threshold to detect the adversary face during the attack. DaD checks the existence of an attack and applies the countermeasures every TIME seconds.

If the CHR is above the CHR threshold, the DaD identifies it as an adversary's face and defines the severity of attack as minor, moderate, or severe. To protect the certificate privacy in VoNDN, DaD was designed based on the following three countermeasures: unpredictable delay, probabilistic cache, and no-cache.

The DaD algorithm is presented in Figure 10. The algorithm uses two attack states (attack detected, no-attack detected), determined by the attack detection method, which is in this case is based on CHR metrics. *i.* **no attack detected**. The default cache replacement policy is applied to all faces, which may be, for instance, LRU. *ii.* **attack detected**. The attack is detected by the detection threshold in the router. The DaD uses, the cache hit ratio to detect the attack. However, other detections mechanisms, such as CRT and name-prefix analyzes can be adapted to the algorithm.

Depending on the severity of the attack, three countermeasures can be applied by DaD to the detected adversary face. Let us consider the application runs at the default phase (no attack detected) and the DaD checks the existence of the attack every TIME seconds. Let us consider that TIME is equal to two seconds for the VoNDN application. If the attack is detected after the check time, the router switches from the default phase to the attack detected phase. In this case, the DaD identifies the attack as minor for 2 seconds and applies a delay phase to the adversary's face. If the attack continues in the next attack check, then DaD considers the attack as moderate and sets the router for the random phase for another 2 seconds to the adversary's face. In the next attack check, if the attack persists, DaD sets the adversary's face of the attacked router for no-cache and keeps always in this state while the attack exists in the next attack checks. If the attack is withdrawn in any phase, then the router returns to the default phase and the whole process restarts again.

TABLE 1. DaD algorithm description and parameters.

inputs	TIME	cacheHitTreshold

auxiliary processes	getCacheHitRatio()
	apply_defaultPolicy()
	apply_Delay()
	apply_Random()
	apply_noCache()	

attack state	ATTACK_DETECTED	delayPhase
		randomPhase
		noCachePhase
	NO_ATTACK_DETECTED	defaultPhase

Table 1 shows the main parameters and the attack phases used by the DaD algorithm. The TIME and the cacheHitTreshold are pre-defined parameters used by this algorithm. TIME is the period used to detect the existence of an attack per face. So, it is the check attack time, which means that every TIME seconds the router is checked for an attack for each face. The router is considered under attack when the cache hit ratio (CHR) is higher than the cacheHitTreshold for a certain face. The auxiliary getCacheHitRatio() process is used to obtain the CHR from each face of the router where DaD is running. If the attack is detected (ATTACK_DETECTED), then distinct countermeasures are applied according to the severity of the attack during the pre-defined time (TIME).

So, to recapitulate the DaD procedure for contents (streamed or certificate), when an attack is firstly detected, the router enters in the delay phase (delayPhase) and keeps in this phase until the next attack check, which will occur TIME seconds later. During the delay phase, all contents are sent through the detected face with an additional unpredictable delay set by the apply_Delay() process. If the attack persists in the second attack check, then the attack detected face enters in the random phase (randomPhase). During this phase, the cached contents are selected from the router's cache with the apply_Random() process to be distributed to the adversary detected face. This random distribution is based on a probabilistic function.

If the attack persists in the third detection period check, then the attack detected face enters the no-cache phase (noCachePhase) and stays in this phase while the attack persists. During this phase, no contents are stored in the cache of the detected face. If the attack is withdrawn in any phase of the ATTACK_DETECTED condition, the router switches to the default phase (defaultPhase).

If the CHR is not above the cacheHitTreshold, then no attack is detected (NO_ATTACK_DETECTED), which establishes the default phase by setting the apply_defaultPolicy() process. In this phase, DaD applies the default caching policy to the router.

The DaD uses the CHR detection metric in privacy-sensitive applications (*e.g.* trusted VoNDN) to identify the adversary node. However, this metric depends on the type of application. Since it is not possible to define a *priori* a threshold for all applications, this may require that the attack check period (TIME) be adjusted automatically by the application. In this way, the DaD algorithm could be adapted to a voice-over NDN application, as discussed next.

C. DaD CONFIGURATION ON VoNDN

The adversary may identify the consumers' private information, such as approximated locations through a targeted certificate in VoNDN. In trusted VoNDN application, the certificates provide integrity for callee and caller. The certificate can be issued by CA or self-signed. However, the cached certificate may provide such pieces of information when it is targeted by the adversary, namely: *i*. the name of callee/caller, *ii*. the location of callee/caller, and *iii*. the approximate time of the conversation.

In VoNDN, the adversary's face is detected if it overcomes the thresholds in 0.2 seconds. Then, multiple countermeasures can be applied by depending on the severity of the attack: *i*. unpredictable delay, *ii*. probabilistic cache, and *iii*. no-cache. In this design, the CHR threshold value can be used to obtain the adversary's face and to identify the severity of the attack. Also, the DaD allows updating the threshold value in each pre-defined time to make decisions about the adversary and countermeasure.

In VoNDN, DaD considers adversary face if a face's CHR up to 1% in best-route and 5% in multicast forwarding strategies. In this application, DaD is analyzing ≈20 packets in every 0.2 seconds for all faces. As presented previously, DaD sets the minor, moderate, and severe phases depending on CHR thresholds these calculated in every 0.2 seconds.

VII. SCENARIO IMPLEMENTATION

In this work, the scenario was implemented and simulated on ndnSIM version 2.6. The simulation scenarios were scripted based on the following: *i*. implement the trusted VoNDN application that simulated in NDN-testbed topology, *ii*. brute-force side-channel timing attack implementation to increase attack success for multiple targets (certificate), and *iii*. DaD privacy model implementation to mitigate the brute-force attack and comparison of the attack mitigation performance with the statically configured countermeasure probabilistic caching.

A. BRUTE-FORCE ATTACK DEVELOPMENT

In traditional side-channel timing attack design, an adversary defines the targets for objects of particular content (segments). However, this type of attack uses to be inefficient in terms of success, when it is directed simultaneously to multiple named certificate.

In this work, an attack design called brute-force was adapted to the side-channel timing attack. Using a brute-force attack, the adversary can target multiple contents and attack them in a short period. This attack can be also considered as a burst attack that is using repeated in short bursts of targets at random intervals [36].

In the brute-force attack, the adversary tries all possible combinations of password dictionaries until getting one that matches [37], [38]. To mitigate this attack, web providers limit the requests for a short period (*e.g.* several failed passwords attempts) and enhancing the complexity of the password dictionary (*e.g.* requiring special characters).

In this work, the brute-force was configured for the content names measuring the response times from the edge router. In this attack, an adversary defines the targets by content segments trying to retrieve them in a short time. The target can be a content segment or a public-key certificate which can be defined by the requirement of the attack. Additionally, this brute-force attack was designed to retrieve streamed content or certificate by randomly and at the same time to increase the attack success.

Attack Procedure: Figure 11 illustrates the brute-force attack process to success the attack for the multiple targets (T_n). These can be the predefined name of content or certificate to start the attack procedure for data packets. If the target (*e.g.* streamed content or certificate) has not been produced by its producer, the NACK packet occurs with "content wasn't available" message.

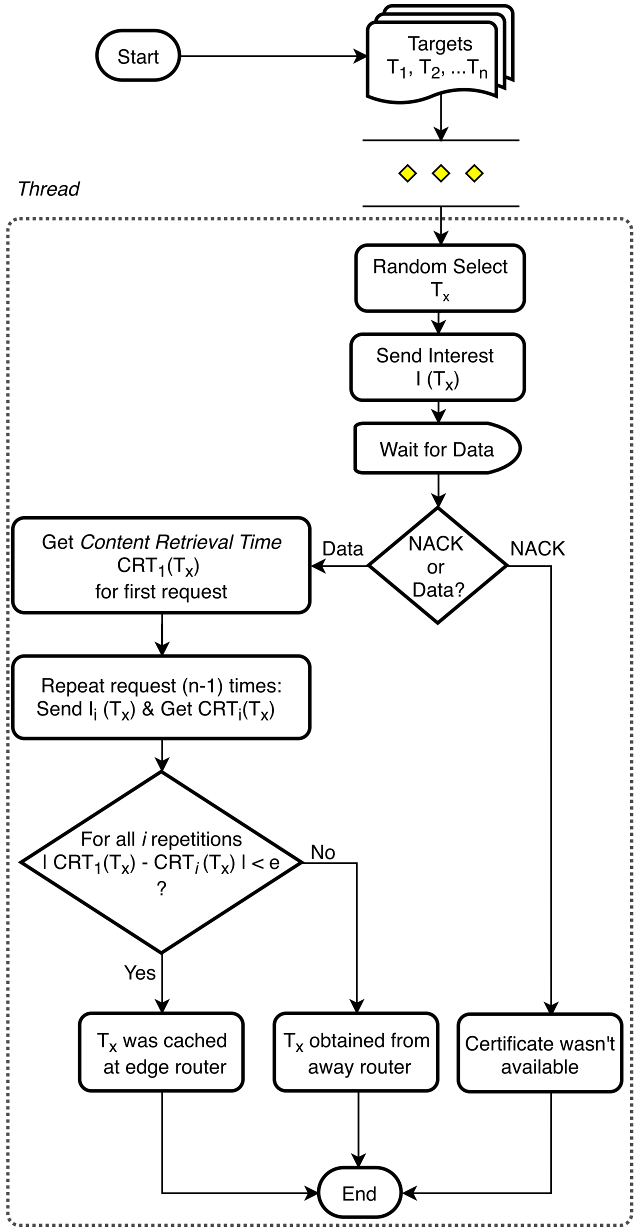


FIGURE 11. Brute-force side-channel timing attack flowchart.

The adversary may repeat the attack several times to distinguish the targets between cached and un-cached. In each repetition, the adversary retrieves different or same CRT values in order to conclude the target is cached by edge or neighbor/away routers. As shown in Figure 11, the adversary selects a target (T_x) randomly from all targets (T_n). To succeed in the attack, the adversary must repeat each of the targets at least two times. In this algorithm, the attack repetition is defined as $n-1$, which can be varied by side-channel attack design. Then a set of CRT values (CRT_i) is obtained for the repetition of a target (T_x).

When the attack finished, the adversary nodes can identify the target location by comparing their CRT. For instance, if the difference between the first CRT and all the others is small (less than ϵ), the CRT_i presents the target (T_x) has been recently cached by the by edge router otherwise it is cached by away routers. Also, the adversary can identify target' geographic distances in terms of hops by analyzing away routers CRT_i as also studied by [13].

Therefore, the attack can be configurable for multiple targets to succeed adversary may able to configure the multiple targets by the brute-force attack. Additionally, the adversary is also able to identify the approximate locations for the missed content(s) by comparing their CRTs.

B. SCENARIO CONFIGURATIONS

In the VoNDN scenario, each content (*e.g.* voice/video, media, and certificate) is signed by its callee and caller to authenticate the conversation. To verify the callee and caller public keys, the certificate authority (CA) publishes a certificate that binds the name and the public-key with a CA signature. The certificates are cached by the NDN-testbed routers to establish the call session in the next time. The trusted VoNDN application scenario was implemented to analyze brute-force attack findings.

1) NDN-TESTBED TOPOLOGY

The VoNDN was simulated on a real NDN-testbed topology. The NDN-testbed is consists of 42 NDN routers on the global participating institutions [8].

Figure 12 illustrates the current global NDN-testbed topology with its gateway routers. This topology was implemented in the NDN simulator (ndnSIM).

TABLE 2. NDN-testbed bandwidth and delays of the links.

Testbed link type	Delay (ms)	Bandwidth (Mbps)
leaf-router	1	1000
router-router	2-155	1000

Table 2 shows the minimum and maximum delays of the links and bandwidths of the NDN-testbed. In testbed, the link delay may vary between the nodes. This caused by a link design between nodes. For instance, minho node is only linked with basel, coruna, copelabs, urjc, and padua. Also, the callee and caller are leaf nodes of the NDN-testbed.

2) VoNDN CONFIGURATION

In the VoNDN application, the callee and caller are paired and send their packets at a constant rate (100 packets/s) to each other. Also, the callee and caller are configured to request a data packet (≈168 kB), that can be presented as voice/video, media, and certificate. These can be requested by an interest packet (≈50 kB). Note that, the voice conversation does not have any silence period because of the callee and callers are configured with a default constant bit rate (CBR). Also, because of VoNDN is implemented only for the attack purposes, encoding/decoding voice and video format (*e.g.* H.264) was not implemented.

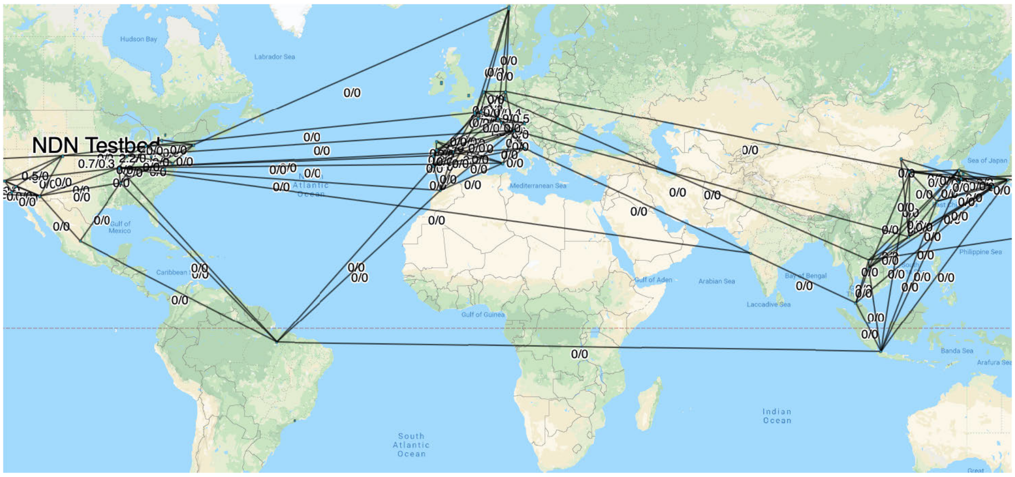


FIGURE 12. NDN-testbed topology (adapted from [8]).

In the VoNDN scenario, each named data packets are signed by a public key to provide the integrity of the callee and caller. This can be trusted by a certificate authority or a self-signed certificate.

Figure 13 illustrates the two-way communication VoNDN application in perspective on how the design of applications and adversaries can be done. In this scenario, the callee and caller publish their interest and data packets to each other without relying on any middle transmission server such as a SIP proxy or a SIP server. To establish a call, the callee and caller exchange data using each-other their unique-call-id (*/vondn/user/unique-call-id*). To authenticate the conversation, the CA publishes certificates (*e.g. /ndn/domain/vondn/KEY*) and these can be cached by gateway routers. Also, the callee or the caller can authenticate themselves by digital signing data packets, using public-key cryptography. For that public-key certificates are required, issued by CA authority.

In a trusted-VoNDN application demonstration (Figure 13), the adversary nodes (*adversary-1* and *adversary-2*) pursues the targets that are recently cached by gateway routers to knowledge Callee (*e.g.* Bob) and Caller (*e.g.* Alice) locations. Each of targeted certificate (*/ndn/domain/vondn/KEY/[cert-name+digest]*), was requested and delivered to the caller or callee. Caller (*caller-1*) public-key certificate is used by the Callee (*callee-1*), while callee-1 public-key certificate is used by the caller-1, to validate the signature and authenticate each-other packets. In this attack, the adversary probes multiple certificates (targets) by brute-force and randomly in order to maximize the success of the attack. Additionally, the attack is repeated at least four times to differentiate the target that has been cached by the edge router or the other routers (neighbor and away). Finally, the adversary can analyze these four CRT samples to conclude that the target has been cached or not by the edge router. Also, other adversaries are (*adversary-*...) attacking to other caller and callee (*callee-*... and *caller-*...) to conclude that their certificates are cached by edge, neighbor, and away routers. For instance, the adversary-1 identifies the following locations: *i*. If the Callee-1 certificate is retrieved from router-1 the target is located at an edge router, *ii*. If the Callee-1 certificate is retrieved from router-2 the target is located at a neighbor router, and *iii* If the Callee-1 certificate is retrieved from router-3 the target is located at away router(s). These decisions are also taken by adversary-2 for router-4, 5, and 6.

On the other hand, the voice/video, media, and certificate packets can be transmitted with different routing strategies to recover the packet loss (*e.g.* due to traffic congestion). The attack scenario was implemented using two routing strategies: *i*. the best route, where packets are routed through the best path between the nodes and *ii*. multicast, where packets are routed to group nodes forwarding strategies. These were implemented with the least recently used (LRU), this cache replacement policy discards the least recently used certificate first from the content store.

3) VoNDN ATTACK SCENARIO CONFIGURATION

Table 3 shows the adversary configuration for the attack scenario. In this model, the selected adversary nodes ($\approx$35%) targeted the certificates (*e.g. /ndn/domain/vondn/KEY/[cert-name+digest]*) that were previously cached by routers and produced by certificate authority (*e.g. /ndn/domain/vondn/KEY*). Each named certificate has an unique sha-256 cryptographic digest function (*e.g. .../sha256digest=fde78cbdff...c4106*) to authenticate the callee and the caller's identity.

The extracted NDN-testbed topology consists of 42 global routers (edge, neighbor, and aways). To establish a VoNDN conversation, an additional 10 consumers (callee and caller) are added to each global router. Therefore, the VoNDN attack scenario is used 462 (420+42) nodes in total and 210 paired VoNDN conversation occurred.

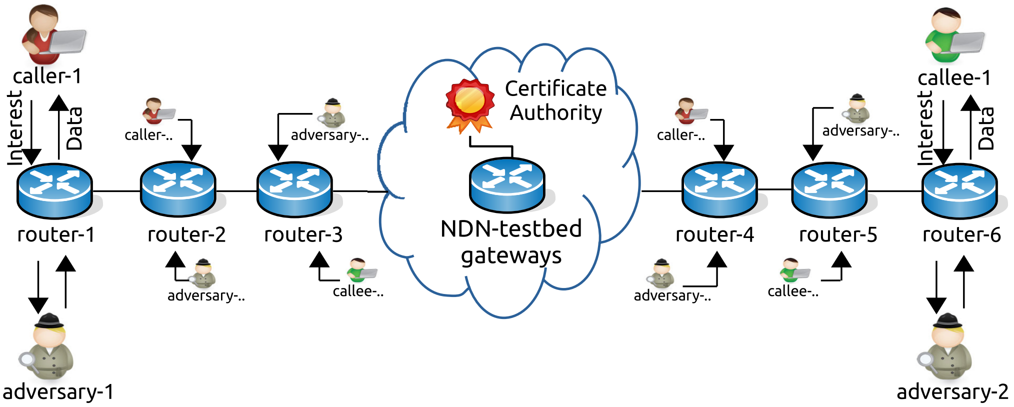


FIGURE 13. VoNDN attack design.

TABLE 3. VoNDN attack scenario configuration.

Network topology	NDN-testbed
Total nodes	462
Attacked edge routers	42 routers
Legitimate nodes	≈65% of total consumer nodes
Adversary nodes	≈35% of total consumer nodes
Certificate authority (CA)	/ndn/domain/KEY/
Targets	/ndn/domain/vondn/KEY/...
Attack repetition	4 for each target
KEY digest	sha256
CS policy	LRU
CS size	1000 packets
CRT decisions	* cached by edge router * cached by neighbor router * cached by away router

In VoNDN, the NDN-testbed nodes that are used to transmit voice/video and certificate packets to the VoNDN consumers via 42 gateway routers. In this scenario, 10 leaf nodes (callee and caller) were assigned to each edge router to represent the callee and the caller. The adversaries attack these 42 edge routers to obtain ≈65% legitimate cached certificate locations.

In the simulation, the adversaries, the callee(s), and the callers were randomly selected for each simulation run. The results were collected from an average of 10 simulation runs for each LRU forwarding strategy (best-route and multicast). The targeted certificates were retrieved by brute-force and each target request was repeated 4 times to increase the attack success. When the attack finished, the adversaries compare the CRTs of certificates that were collected from the routers to decide where/when the certificates have been cached. As shown in Table 3, the NDN-testbed routers were classified by the following terms: *i*. *edge routers* represent the first-hop routers of the leaf nodes, *ii*. *neighbor routers* are the second hop routers of the leaf nodes, and *iii*. *away routers* are those located at more than two hops away from the leaf nodes. A certificate authority may be located in an away router.

In these experiments, the CHR, CRT, and hop count metrics were used to analyze the attack behaviors to distinguish between legitimate and adversarial requests. These results are presented next.

VIII. RESULTS

To analyze the simulated scenario results, the metrics CRT, CHR, and hop counts were analyzed for *best-route* and *multicast* forwarding strategies. The results were analyzed based on the following: *i*. to evaluate the brute-force attack performance for multiple targets in trusted VoNDN application using the CHR results, *ii*. to analyze the attack based on the location information about the callee and the caller, *iii*. to compare DaD performance with a static countermeasure (probabilistic caching) to mitigate the brute-force timing attack, and *iv*. to analyze the performance of the content distribution between a statically configured countermeasure (probabilistically caching) and DaD by analyzing the CRT and hop count metrics.

In this setup, the probabilistically caching was implemented that stands as a static router configuration to be compared with DaD. This comparison attempts to show that how DaD can be an efficient approach to mitigate the attack and maintaining the legitimate certificate requests in the VoNDN testing scenario.

A. ATTACK PERFORMANCE EVALUATION

With multicast forwarding strategy, multiple paths are followed by data packets, and contents are cached in more routers, while in best-route only the best path's routers cache the content. In these experiments, the adversaries can distinguish between cached and un-cached targets through the retrieved CRT values and take an attack decision about the callee and the caller locations.

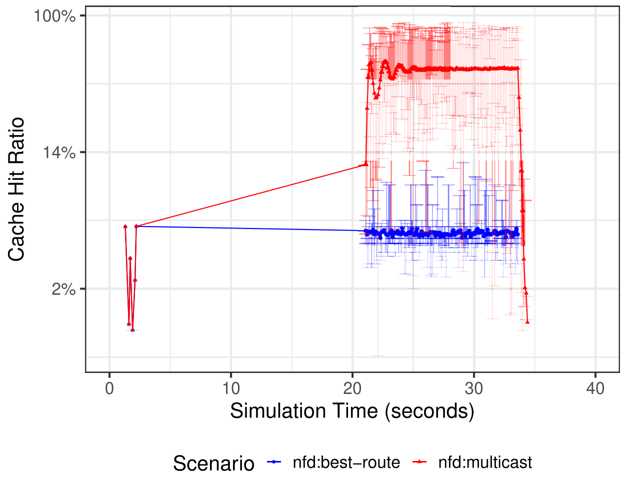


FIGURE 14. VoNDN brute-force attack performance on forwarding strategies.

Figure 14 illustrates the results of the CHR values to measure the performance of the attack for best-route and multicast forwarding strategies with LRU configuration on NDN-testbed topology. In this scenario, the certificates were previously cached by edge routers which were used to establish a voice/video conversation (≈0-5 sec.). Thus, the average of CHR was calculated globally based on all edge routers, as defined by previous Eq. 3. The attack period occurs in the interval of ≈21-40 seconds.

The adversaries targeted the certificates to know the location of callee and caller (≈65% of legitimate nodes). To improve the success of the attack, the brute-force procedure can be repeated by an adversary. By accomplishing this, an adversary can distinguish between first and last repetitions. In this attack scenario, the adversary nodes retrieve the targets with 4 repetitions (Table 3). Thirty-five percent of adversary nodes were able to target 252 certificates to identify the locations of legitimate nodes. The attack performance measured in terms of global CHR, given by Eq. 3, presented the following values for the edge routers: ≈1% for nfd:best-route and ≈50.5% for nfd:multicast.

B. CERTIFICATE LOCATION DETERMINATION

The public key certificates are cached by the NDN-testbed routers. Since the adversary targets the consumers' certificates, these can identify the location of the consumer.

If the target (certificate) has been cached in the edge router, then the adversary hits the cache and obtains the minimum CRT. Through this attack, an adversary can identify the targets that have been cached by the edge router. Moreover, the adversary can determine the un-cached target locations by analyzing the CRT values. For instance, the maximum CRT reveals that the certificate has not been cached by any router, except by its producer (CA). If the CRT obtained is between minimum and maximum, an adversary concludes that the target has been cached by neighbor routers.

In this experiment, the adversaries were configured to distinguish the location of the cached and un-cached certificate by comparing each of the collected CRTs. The CRT values are used to classify the targets, based on three locations: *i.* cached by edge routers, *ii.* cached by neighbor routers, and *iii.* cached by away routers.

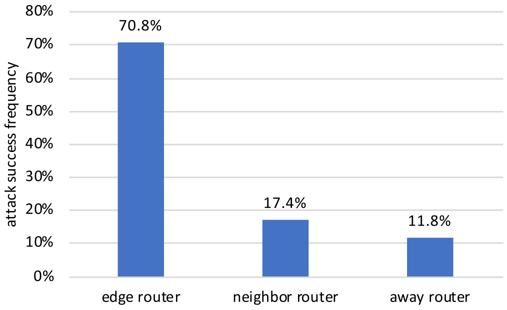


FIGURE 15. Determine certificate locations in VoNDN.

Figure 15 shows the results of the target locations based on the CRTs obtained in the VoNDN multicast default scenario without countermeasures. In this experiment, the adversaries were configured to target all certificates these are cached by various locations. The computed location findings are based on all adversary's CRT calculations. The adversaries concluded that the targets were located on the routers as follows: *i.* 70.8% cached by the edge router, *ii.* 17.4% cached by the neighbor router, and *iii.* 11.8% cached by the away routers or certificate authority. In this attack scenario, the adversaries are configured to attack all possible targets at the same time. Because of this, the success of the attack is computed highest of the edge router. Also, the forwarding strategy increased to the success of the attack because every node cached the certificates with multicast.

C. COUNTERMEASURES

To mitigate the attack, the countermeasures based on static probabilistic and DaD were configured with the NDN forwarding daemon (NFD), which used as a network forwarder. The countermeasures were implemented in VoNDN with the best-route and the multicast forwarding strategies on the default LRU scenario.

The following configurations were used to mitigate the brute-force attack:

1) The edge routers were statically configured with a probabilistic caching of 10% of content cache acceptance by randomly chosen of the data packet that can be cached. The global CHR results (adversary's faces) of these edge routers were analyzed and compared with the LRU, which is used replacement policy in ndnSIM.
2) The edge routers were configured within a DaD algorithm, which identifies the face of the router that is being attacked by checking the CHR threshold every 0.2 seconds and applying each countermeasures phase

during 2 seconds. When an attacked face is detected, the DaD applies countermeasure strategies, depending on the severity of the attack. To detect the face that is being attacked and apply countermeasures, the CHR threshold values were used. In this work, the threshold values were identified only for this particular attack scenario which may be different on other NDN applications. Through the VoNDN simulation experiences, a predefined CHR threshold was identified by moving average values as 1% CHR for best-route and 5% CHR for multicast forwarding strategies. If the attack was withdrawn by an adversary or does not exist, the DaD applies the default (LRU) phase.

In VoNDN, the implemented DaD checks the existence of an attack on the faces every 0.2 seconds and the countermeasure phases (each for 2 sec.) are only applied to the attacked edge routers, in order to protect the legitimate certificate requests from the edge router(s). The DaD detects the attack by checking the CHR (Eq.5 with a α=0) threshold on every face.

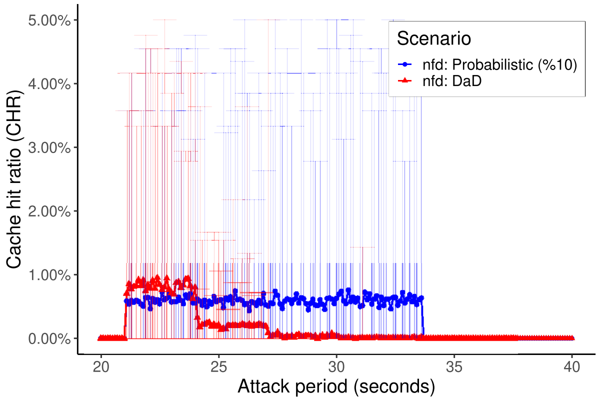


FIGURE 16. Comparisons of applied countermeasures in VoNDN best-route forwarding strategy.

1) APPLIED COUNTERMEASURES ON THE VoNDN

Figure 16 illustrates the VoNDN CHR results obtained with a brute-force attack, considering the use of the probabilistic caching (10%) and the DaD in the routers. In both cases, the best-route forwarding strategy was used. An average CHR of 0.69% was obtained in the attack period with the probabilistic caching, which mitigated the attack ≈30.3% when compared with the results of the default LRU scenario (Figure 14). On the other hand, the DaD detects the attacked router first then applies different countermeasures phases while the attack persists with a 1% CHR threshold. If no attack is detected, then DaD sets the router face to the default phase. The average CHR obtained was 0.42%, which represents attack mitigation of ≈57.5%, when compared to the default LRU best-route scenario. The DaD also performed 39.1% improved the attack mitigation compared to the probabilistic caching configuration.

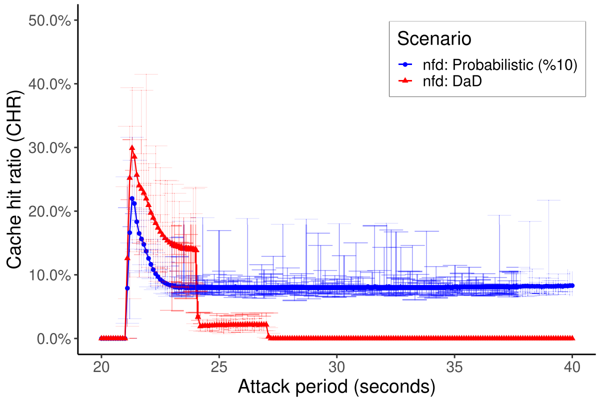


FIGURE 17. Comparisons of applied countermeasures in VoNDN multicast forwarding strategy.

Figure 17 shows the CHR results obtained in a multicast forwarding strategy in VoNDN. The discontinuities seen at 21s, 24.5 s, and 27.5 s in the DaD graphics of Figure 17 (and Figure 16) are due to the application of the countermeasures. Using the probabilistic caching (10%) for all faces, an average CHR of 8.12% was obtained during the attack period. This configuration mitigated the attack about 83.9% when compared with the default LRU scenario in multicast (Figure 14). The CHR threshold in DaD was configured to 5% for a multicast attack scenario. In this case, an average of CHR 5.15% was obtained, which mitigated the attack 89.8%, when compared to the default LRU multicast scenario (Figure 14), and mitigated the attack 36.5%, when compared with the probabilistic caching.

2) COUNTERMEASURES DISTRIBUTION EFFICIENCY EVALUATION

The DaD only applies the countermeasures to the attack detected faces instead of setting countermeasures to all faces. Thus, legitimate requests and privacy can be preserved by the DaD. To show this, the CRT (best-route) and hop counts (multicast) were analyzed on default (LRU) and countermeasures (Probabilistic and DaD) VoNDN scenarios.

3) CRT RESULTS

Figure 18 illustrates the global CRT results (best-route) for both adversary and legitimate nodes considering on default LRU (best-route), probabilistic caching, and DaD scenarios during the attack time (21-40 s). If the target is cached by the edge router, the minimum CRT values are obtained for adversary nodes otherwise it obtains increased CRT values for neighbor and away targets. To show the countermeasures (probabilistic and DaD) results to mitigate the attack on the default (LRU) scenario, the CRT values were analyzed. When the probabilistic and the DaD were applied, the adversary node's CRT value increases for the targets which are illustrating the attack mitigation rate. In this case, the adversary may not able to identify the location of the cached target because of unsteady collected CRTs.

The countermeasures (probabilistic and DaD) can be used to mitigate the attack. However, the CRT results showing that the static probabilistic caching also increases the legitimate node CRTs, which reduces the content distribution performance for the legitimate nodes. To preserve the legitimate nodes' requests, the DaD applies the countermeasures only to the face that being attacked. Thus, an average of CRT is calculated as the same (0.056 ms) for nfd:DaD and nfd:LRU (best-route).

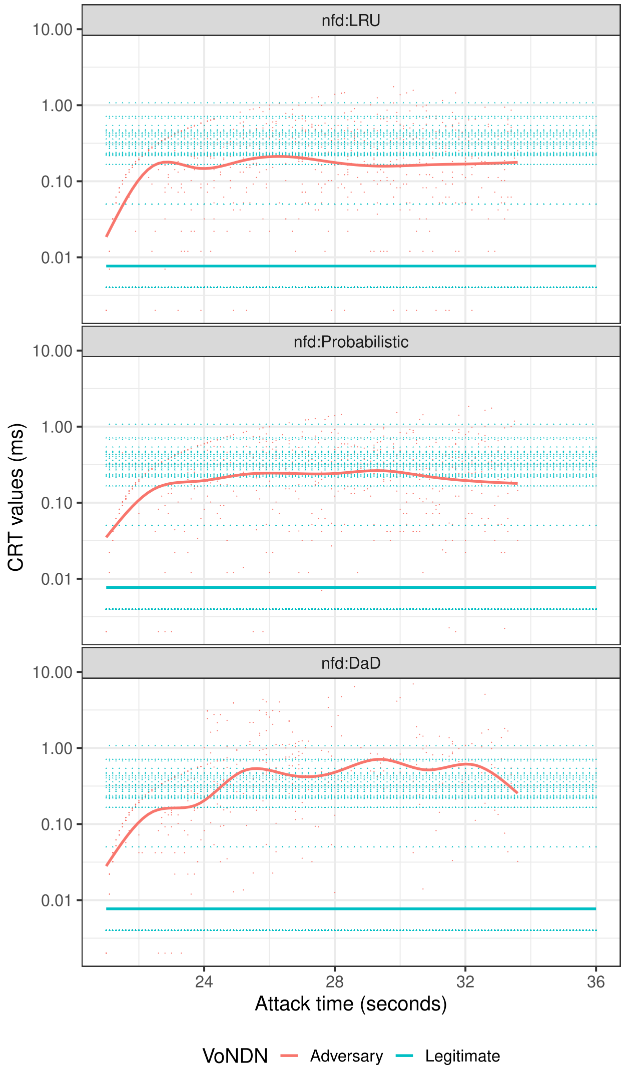


FIGURE 18. VoNDN Global CRT results for the adversary and legitimate nodes.

Table 4 shows, the CRT metrics these illustrated in Figure 18 in VoNDN. During the attack period, the legitimate CRT metrics were preserved by the DaD, compared to the probabilistic caching. Also, the CRT values of the adversaries are higher than those of the legitimate nodes because the adversaries were also targeted to un-cached certificates in order to obtain neighbor and away routers. Because DaD only applies the countermeasures to the adversary's faces, the average CRT of DaD calculated 0.419 ms which is higher than CRT of probabilistic (0.328 ms). Thus, DaD mitigated more attack than static probabilistic countermeasure while protecting legitimate requests.

TABLE 4. VoNDN CRT average values for legitimate and adversary nodes.

scenarios	CRT average (ms)	
	legitimate	adversary
LRU (best-route)	0.056	0.270
probabilistic	0.093	0.328
DaD	0.056	0.419

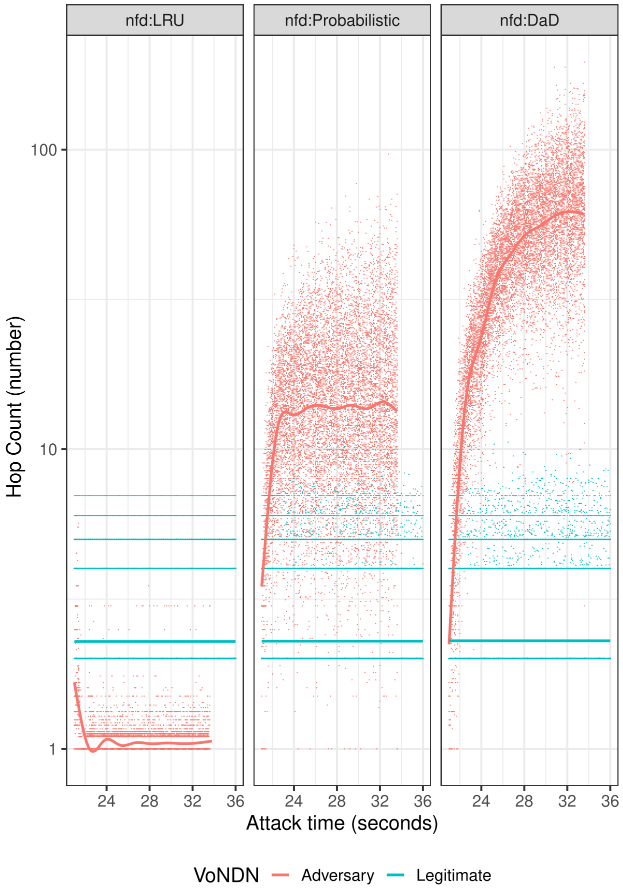


FIGURE 19. VoNDN adversary and legitimate nodes hop count metrics.

4) HOP COUNT RESULTS

Figure 19 shows the global hop-count results on default LRU (multicast), probabilistic, and DaD scenarios during the attack time (21-40 s). If the attack became succeed the minimum hop count metric can be obtained because of adversary hits the edge routers. If an attack is not successful, the maximum hop count metrics obtained. Because of no-countermeasure applied in LRU (multicast) scenario, the adversary's average of hop-count metrics is obtained minimum (≈1.04) than the average of legitimate node's hop counts (≈2.38) during the attack.

To mitigate the attack on the default (LRU) scenario, the countermeasures (probabilistic and DaD) were set. When countermeasures are applied, the hop count of adversaries increases. However, the hop counts of legitimate nodes also increase in probabilistic caching configuration, because of its set to all faces. On the other hand, the DaD only set the countermeasures to attack the detected face, which preserved the hop counts of legitimate requests.

TABLE 5. VoNDN hop count countermeasure results.

scenarios	hop count (average)	
	legitimate	adversary
LRU (multicast)	2.38	1.04
probabilistic	2.43	1.84
DaD	2.38	35.90

Table 5 shows the average hop count metrics for the applied countermeasures (probabilistic and DaD) to mitigate the attacks on the default multicast (LRU) VoNDN scenario. During the attack period (21-40 s), the default scenario presented an average hop count of 2.38 for legitimate nodes and 1.04 for the adversary nodes.

The probabilistic and DaD scenarios are applied in order to mitigate the attack on the default scenario. In the probabilistic scenario, the average of adversaries hop count increased to 1.84 from 1.04 (default). This reveals the attack mitigation of about 55% on the default scenario. However, the probabilistic caching also increased the average hop counts of the legitimate certificate requests from 2.38 (default) to 2.43. Therefore, 2% of the VoNDN conversation traffic between the callee and the caller may be considered as affected or delayed because of the probabilistic scenario.

In DaD, the legitimate requests preserved and the average hop count was equal to the value obtained in the LRU (default) scenario (2.38). These values suggest that DaD may have applied the countermeasures only to the attack detected faces. In DaD, the average hop count was increased to 35.9 from 1.04 for the adversary nodes. These results are showing that the performance of attack is significantly decreased under DaD's multiple countermeasure configuration.

IX. CONCLUSION AND FUTURE WORK

In this work, a brute-force side-channel timing attack and a countermeasure approach called DaD were presented for in-network caching based trusted VoNDN application. The attack scenario was simulated on NDN-testbed topology. The adversaries targeted 252 certificates cached in the NDN-testbed routers to gather information for the callee and caller locations.

To increase the success of the attack, a brute-force attack was designed based on a random selection of multiple targets. The adversaries succeeded to determine the location of the targets by the following values: 70.8% cached by the edge routers, 17.4% cached by the neighbor routers, and 11.8% cached by away routers. Also, the performance of brute-force attack implementation in terms of the CHR on the edge routers was the following: 1% CHR in best-route, and 50.5% CHR in multicast forwarding strategies with default (LRU) scenario. To mitigate the brute-force attack on the default scenario, the probabilistic caching and DaD were implemented. In terms of CHR, the results showed that the attack mitigation improved with the DaD when compared with the probabilistic caching, by the following values: 39.1% in best-route and 36.5% in the multicast forwarding strategies.

In terms of average CRT, the results show that the attack mitigation with DaD improved 51% when compared with the probabilistic caching. Moreover, DaD obtained an average CRT of the legitimate nodes equal to the value obtained with the default (LRU), which suggests that the legitimate requests were preserved. Unlike the probabilistic caching, the average hop count results show that the DaD only applies the countermeasures to the attack detected faces, which helps to preserve the legitimate node requests. Also in terms of hop counts, the use of probabilistic caching revealed that 2% of the legitimate requests were not preserved when compared with the DaD.

FUTURE WORK

In this work, the CHR thresholds and the detection period were identified manually. In the future, the threshold and the detection period will be adjusted dynamically using machine learning techniques.

SOURCE CODE

Scenarios were scripted by the C++11 library in ndnSIM 2.6. The scenario implementations and required tools can be publicly accessible at the author's GitHub account—https://git.io/JJqEw

REFERENCES

[1] L. Zhang, D. Estrin, J. Burke, V. Jacobson, J. D. Thornton, D. K. Smetters, B. Zhang, G. Tsudik, D. Massey, C. Papadopoulos, L. Wang, P. Crowley, and E. Yeh, "Named data networking (NDN) project," NDN, Xerox Palo Alto Res. Center-PARC, Palo Alto, CA, USA, Tech. Rep. NDN-0001, 2010. [Online]. Available: http://named-data.net/techreport/TR001ndn-proj.pdf

[2] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, "Networking named content," in *Proc. 5th Int. Conf. Emerg. Netw. Exp. Technol. (CoNEXT)*. New York, New York, USA: ACM, vol. 30, no. 2, 2009, p. 1, doi: 10.1145/1658939.1658941.

[3] A. Compagno, M. Conti, E. Losiouk, G. Tsudik, and S. Valle, "A proactive cache privacy attack on NDN," in *Proc. NOMS-IEEE/IFIP Netw. Oper. Manage. Symp.*, Apr. 2020, pp. 1–7, doi: 10.1109/NOMS47738.2020.9110318.

[4] G. Acs, M. Conti, P. Gasti, C. Ghali, and G. Tsudik, "Cache privacy in named-data networking," in *Proc. IEEE 33rd Int. Conf. Distrib. Comput. Syst.*, Jul. 2013, pp. 41–51, doi: 10.1109/ICDCS.2013.12.

[5] E. W. Felten and M. A. Schneider, "Timing attacks on Web privacy," in *Proc. 7th ACM Conf. Comput. Commun. Secur. (CCS)*, 2000, pp. 25–32, doi: 10.1145/352600.352606.

[6] A. Chaabane, E. De Cristofaro, M.-A. Kaafar, and E. Uzun, "Privacy in content-oriented networking: Threats and countermeasures," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 43, no. 3, pp. 26–33, 2012, doi: 10.1145/2500098.2500102.

[7] S. Di Benedetto, P. Gasti, G. Tsudik, and E. Uzun, "ANDaNA: Anonymous named data networking application," in *Proc. Netw. Distrib. Syst. Secur. Symp.*, 2011, pp. 1–18. [Online]. Available: http://arxiv.org/abs/1112.2205

[8] (2015). *NDN Testbed-Named Data Networking (NDN)*. [Online]. Available: https://named-data.net/ndn-testbed/

[9] T. Lauinger, "Security & scalability of content-centric networking," TU Darmstadt, Darmstadt, Germany, Tech. Rep. 2275, 2010, p. 60. [Online]. Available: http://tuprints.ulb.tu-darmstadt.de/2275/1/ccn-thesis.pdf

[10] A. Mohaisen, H. Mekky, X. Zhang, H. Xie, and Y. Kim, "Timing attacks on access privacy in information centric networks and countermeasures," *IEEE Trans. Dependable Secure Comput.*, vol. 12, no. 6, pp. 675–687, Nov. 2015, doi: 10.1109/TDSC.2014.2382592.

[11] T. Lauinger, N. Laoutaris, P. Rodriguez, T. Strufe, E. Biersack, and E. Kirda, "Privacy risks in named data networking: What is the cost of performance?" *ACM SIGCOMM Comput. Commun. Rev.*, vol. 42, no. 5, pp. 54–57, Sep. 2012, doi: 10.1145/2378956.2378966.

[12] T. Lauinger, N. Laoutaris, and P. V. Rodriguez, "Privacy implications of ubiquitous caching in named data networking architectures," *ACM Sigcomm*, vol. 42, no. 5, pp. 54–57, 2012. [Online]. Available: https://tobias.lauinger.name/papers/ccn-cache-attacks-tr-iseclab-0812-001.pdf

[13] A. Compagno, M. Conti, P. Gasti, L. V. Mancini, and G. Tsudik, "Violating consumer anonymity: Geo-locating nodes in named data networking," in *Applied Cryptography and Network Security* (Lecture Notes in Computer Science: Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 9092. Cham, Switzerland: Springer, 2015, pp. 243–262, doi: 10.1007/978-3-319-28166-7_12.

[14] S. K. Fayazbakhsh, Y. Lin, A. Tootoonchian, A. Ghodsi, T. Koponen, B. Maggs, K. C. Ng, V. Sekar, and S. Shenker, "Less pain, most of the gain: Incrementally deployable ICN," in *Proc. ACM SIGCOMM Conf. SIGCOMM (SIGCOMM)*, 2013, vol. 43, no. 4, p. 147. [Online]. Available: http://dl.acm.org/citation.cfm?id=2486001.2486023

[15] C. Bernardini, S. Marchal, M. R. Asghar, and B. Crispo, "PrivICN: Privacy-preserving content retrieval in information-centric networking," *Comput. Netw.*, vol. 149, pp. 13–28, Feb. 2019, doi: 10.1016/j.comnet.2018.11.012.

[16] D. Kondo, T. Silverston, V. Vassiliades, H. Tode, and T. Asami, "Name filter: A countermeasure against information leakage attacks in named data networking," *IEEE Access*, vol. 6, pp. 65151–65170, 2018, doi: 10.1109/ACCESS.2018.2877792.

[17] D. Chaum and E. Van Heyst, "Group signatures," in *Advances in Cryptology—EUROCRYPT* (Lecture Notes in Computer Science: Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 547, no. 2. Berlin, Germany: Springer, 1991, pp. 257–265, doi: 10.1007/3-540-46416-6_22.

[18] D. Boneh, X. Boyen, and H. Shacham, "Short group signatures," in *Advances in Cryptology—EUROCRYPT* (Lecture Notes in Computer Science: Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 3152. Berlin, Germany: Springer, 2004, pp. 41–55, doi: 10.1007/978-3-540-28628-8_3.

[19] L. Zhang, A. Afanasyev, J. Burke, V. Jacobson, K. Claffy, P. Crowley, C. Papadopoulos, L. Wang, and B. Zhang, "Named data networking," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 3, pp. 66–73, Jul. 2014, doi: 10.1145/2656877.2656887.

[20] C. Yi, A. Afanasyev, L. Wang, B. Zhang, and L. Zhang, "Adaptive forwarding in named data networking," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 42, no. 3, p. 62, 2012, doi: 10.1145/2317307.2317319.

[21] Y. Yu, "Public key management in named data networking," NDN, UCLA, Los Angeles, CA, USA, Tech. Rep. NDN-0029, 2015. [Online]. Available: http://named-data.net/techreports.html

[22] D. Van Jacobson, M. Stewart, J. Thornton, and R. Braynard, "VoCCN: Voice over content-centric networks," in *Proc. ReArch*, 2009, p. 1, doi: 10.1145/1658978.1658980.

[23] R. Birke, M. Mellia, M. Petracca, and D. Rossi, "Experiences of VoIP traffic monitoring in a commercial ISP," *Int. J. Netw. Manage.*, vol. 20, no. 5, pp. 339–359, Aug. 2010, doi: 10.1002/nem.758.

[24] G. Zhang, S. Fischer-Huebner, L. A. Martucci, and S. Ehlert, "Revealing the calling history of SIP VoIP systems by timing attacks," in *Proc. Int. Conf. Availability, Rel. Secur.*, 2009, pp. 135–142, doi: 10.1109/ARES.2009.129.

[25] C. V. Wright, L. Ballard, S. E. Coull, F. Monrose, and G. M. Masson, "Spot me if you can: Uncovering spoken phrases in encrypted VoIP conversations," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2008, vol. 149, no. 7, pp. 35–49. [Online]. Available: http://ieeexplore.ieee.org/document/4531143/

[26] S. Chen, R. Wang, X. Wang, and K. Zhang, "Side-channel leaks in Web applications: A reality today, a challenge tomorrow," in *Proc. IEEE Symp. Secur. Privacy*, May 2010, pp. 191–206, doi: 10.1109/SP.2010.20.

[27] J. Peterson and C. Jennings, "Enhancements for authenticated identity management in the session initiation protocol (SIP)," Internet Soc., Reston, VA, USA, Tech. Rep. rfc4474, 2006. [Online]. Available: https://tools.ietf.org/pdf/rfc4474.pdf

[28] S. Schinzel, "An efficient mitigation method for timing side channels on the Web," in *Proc. 2nd Int. Workshop Constructive Side-Channel Anal. Secure Design (COSADE)*, 2011, pp. 1–6.

[29] I. Psaras, W. K. Chai, and G. Pavlou, "Probabilistic in-network caching for information-centric networks," in *Proc. 2nd Ed. ICN Workshop Inf.-Centric Netw. (ICN)*. New York, NY, USA: ACM, 2012, p. 55, doi: 10.1145/2342488.2342501.

[30] R. Wiangsripanawan, W. Susilo, and R. Safavi-Naini, "Design principles for low latency anonymous network systems secure against timing attacks," in *Proc. Conf. Res. Pract. Inf. Technol. Ser.*, vol. 68, Jan. 2007, pp. 183–191. [Online]. Available: https://dl.acm.org/doi/10.5555/1274531.1274553

[31] L. Yao, Z. Fan, J. Deng, X. Fan, and G. Wu, "Detection and defense of cache pollution attacks using clustering in named data networks," *IEEE Trans. Dependable Secure Comput.*, early access, Oct. 16, 2018, doi: 10.1109/TDSC.2018.2876257.

[32] M. Conti, P. Gasti, and M. Teoli, "A lightweight mechanism for detection of cache pollution attacks in named data networking," *Comput. Netw.*, vol. 57, no. 16, pp. 3178–3191, Nov. 2013, doi: 10.1016/j.comnet.2013.07.034.

[33] E. Dogruluk, A. Costa, and J. Macedo, "A detection and defense approach for content privacy in named data network," in *Proc. 10th IFIP Int. Conf. New Technol., Mobility Secur. (NTMS)*, Jun. 2019, pp. 1–5, doi: 10.1109/NTMS.2019.8763835.

[34] E. Dogruluk, A. Costa, and J. Macedo, "Identifying previously requested content by side-channel timing attack in NDN," in *Future Network Systems and Security* (Communications in Computer and Information Science), vol. 878. Cham, Switzerland: Springer, Aug. 2018, pp. 33–46, doi: 10.1007/978-3-319-94421-0_3.

[35] E. Dogruluk, A. Costa, and J. Macedo, "Evaluating privacy attacks in named data network," in *Proc. IEEE Symp. Comput. Commun. (ISCC)*, Jun. 2016, pp. 1251–1256, doi: 10.1109/ISCC.2016.7543908.

[36] T. N. D. Pham, C. K. Yeo, N. Yanai, and T. Fujiwara, "Detecting flooding attack and accommodating burst traffic in delay-tolerant networks," *IEEE Trans. Veh. Technol.*, vol. 67, no. 1, pp. 795–808, Jan. 2018, doi: 10.1109/TVT.2017.2748345.

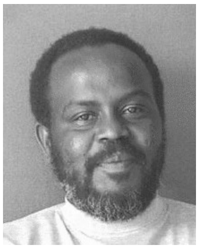
[37] A. Hayes, "Network service authentication timing attacks," *IEEE Secur. Privacy*, vol. 11, no. 2, pp. 80–82, Mar. 2013, doi: 10.1109/MSP.2013.39.

[38] J. Owens and J. Matthews, "A study of passwords and methods used in brute-force SSH attacks," Clarkson Univ., Potsdam, NY, USA, Tech. Rep., 2008. [Online]. Available: https://people.clarkson.edu/~owensjp/pubs/leet08.pdf

ERTUGRUL DOGRULUK (Student Member, IEEE) received the M.Sc. degree in telecommunications engineering from the University of Sunderland, U.K. He is currently pursuing the Ph.D. degree with the Department of Informatics, University of Minho, Portugal. He is currently a Researcher with the Computer Communications and Networks Laboratory, Centro Algoritmi Research Center. He also works as a Teaching Assistant with the University of Minho. His main research interest includes cache privacy in next-generation computer networks.

ÓSCAR GAMA received the Ph.D. degree in electronics and computers from the University of Minho, in 2011. He has been working as a Researcher with the University of Minho, since 2012. He has joined several research projects, mainly focused in sensors, protocols, and currently in ITS simulation environments for vehicular ad hoc networks, in the scope of the CAR2X Communications Project.

ANTÓNIO D. COSTA (Member, IEEE) received the degree in systems and informatics engineering, the M.Sc. degree in informatics, and the Ph.D. degree in computer science from the University of Minho, Portugal, in 1992, 1998, and 2006, respectively. He is currently an Assistant Professor with the Department of Informatics, University of Minho, where he develops teaching and research activities in the fields of computer networks and computer communications, since 1992. As a Researcher, he also integrates the Centro Algoritmi, Computer Communications and Networks (CCN) Research Group, University of Minho.



JOAQUIM MACEDO received the degree in electrical engineering, telecommunications, and electronics and the degree in computer science from Agostinho Neto University, Angola, in 1983 and 1985, respectively, and the Ph.D. degree in computer engineering from the University of Minho, Portugal, in 2002. He has been an Assistant Professor with the Department of Informatics, School of Engineering, University of Minho, since 2002. He develops his research activity as an Integrated Member with the Algoritmi Center, University of Minho. His teaching and research interests include information retrieval, computer networks, and the use of ICTs in developing countries. In this context, he has participated in several research projects, supervising master and doctoral students. He has more than four dozen peer-reviewed papers in journals and scientific conferences. He collaborated for more than ten years with the Catholic University of Angola, as a Visiting Professor.