Producer Anonymity based on Onion Routing in Named Data Networking

ICNRG Interim Meeting,

1 December, 2020

Kentaro Kita, Yuki Koizumi, Toru Hasegawa,
Onur Ascigil, Ioannis Psaras

Kentaro Kita, Yuki Koizumi, Toru Hasegawa, Onur Ascigil and Ioannis Psaras, "Producer Anonymity based on Onion Routing in Named Data Networking," IEEE Transactions on Network and Service Management, Aug. 2020.

When Anonymity is Needed

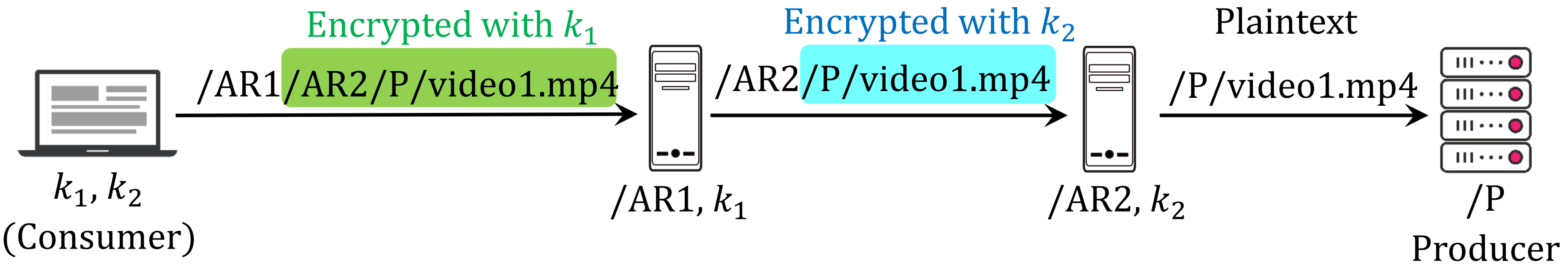
- Definition
  - Consumer anonymity : content-consumer unlinkability
    - Adversaries cannot learn who requests some specific content
  - Producer anonymity : content-producer unlinkability
    - Adversaries cannot learn who publishes some specific content
- Usage Scenario
  - Privacy-sensitive applications/protocols
    - Location-based service
    - Application that deals with health information of users
      - E.g.) Assume that Bob agrees to offer his health information, such as his age, weight, and blood pressure value, to a server for statistical surveys. However, he might wish to hide his identity from the server for his privacy.
  - Censorship evasion
    - E.g.) Assuming that Alice wishes to launch a website that provides people with information about fraud by some companies or governments, she may lose her job or be punished if she is not anonymous

Existing Studies

- Consumer anonymity
    - Inspired by onion routing-based systems in IP
        - ANDaNA [1] : Initial attempt to adapt Tor to NDN
            - Briefly explained in the following slides
    - Inspired by P2P-based anonymity systems in IP
        - CRISP [2]
            - To prevent adversaries to trace back an Interest packet to its origin, each router probabilistically determines whether to forward a received Interest packet toward the specified producer or toward another cooperative router
- Producer anonymity
    - NDN-ABS (NDN Attribute-based Signature) [3]
        - Signatures are generated so that consumers cannot identify a single producer among a set of producers with the same attribute
        - NDN-ABS addresses information leakage only from signatures but this is insufficient to completely achieve producer anonymity (explained later)

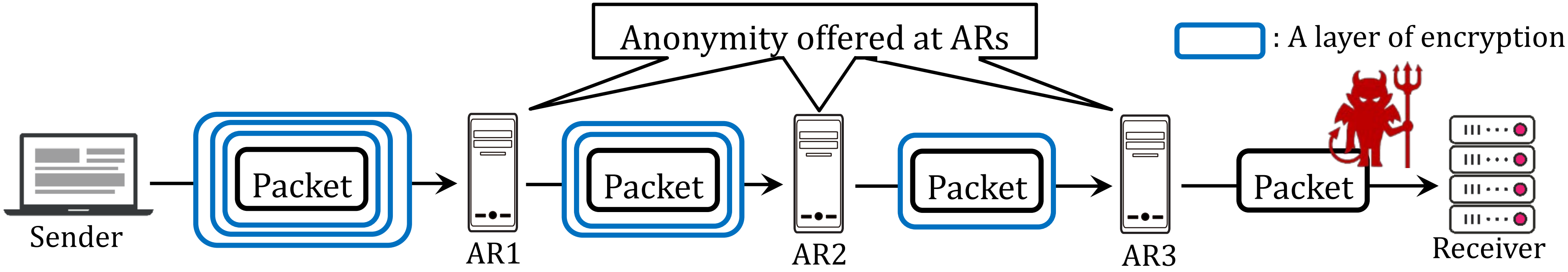
ANDaNA

- Threat
  - Adversaries who eavesdrop packets on compromised network entities to trace their origins
- System Overview
  - A consumer chooses a series of two **anonymizing routers (ARs)**, called a **circuit**, and exchange secret keys (k_1, k_2)
    - AR : A voluntary server on which the ANDaNA application is installed
  - The consumer issues Interest packets whose name is encapsulated in multiple layers of secret key encryption along the circuit
  - Each AR decrypts the top layer and forwards it to the next hop
  - (Data packets are returned in the opposite direction from a router's cache or the producer while being encrypted)

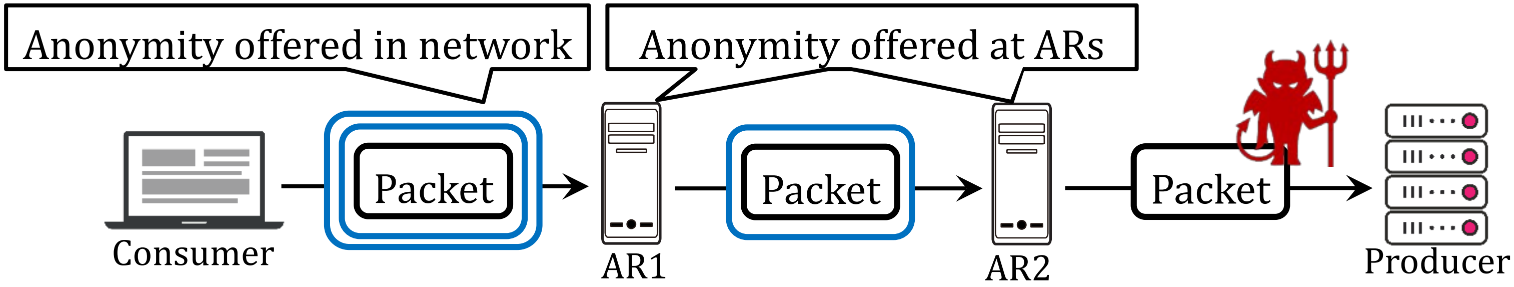


ANDaNA vs Tor in IP

- Advantage of ANDaNA
  - It achieves a level of anonymity comparable to Tor with one fewer ARs
- Comparison
  - With Tor, anonymity is offered only at ARs
    - Because each packet is forwarded while altering its bit pattern by decryption, adversaries cannot trace its origin



  - With ANDaNA, anonymity is offered in network and at ARs
    - Anonymity is naturally achieved because **Interest packets do not carry information on consumers**



Threats to Producer Anonymity

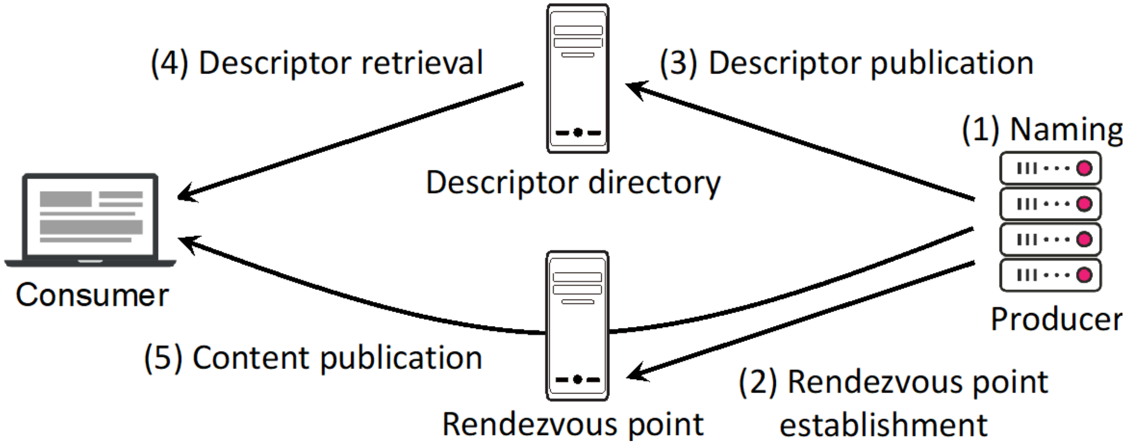
- Adversaries can correlate content and its producer by using
  - **Content name** and **signature**
    - The bindings between (producer identity, name, public key) are established to enable consumers to verify the provenance of content [4]
      - Human-readable name binds producer identity and name
        - E.g.) /CNN/Alice/video1.mp4
      - Public key certificate binds producer identity and public key, and name and public key [5]
        - Consumers trusts certificate authorities to publish certificates only to identity confirmed producers
        - Certificate name is managed under the producer's namespace
  - **Packet route**
    - By eavesdropping packets on compromised network entities, adversaries can identify who publishes what content
  - (We do not consider information leakage from content payload)
    - This must be managed by each producer, not the system

Goal and Approach

- Goal
  - Design a system that achieves producer anonymity against adversaries who leverage content names, signatures, and packet route
  - The system achieves producer anonymity efficiently by taking advantage of NDN (like ANDaNA)
- Approach
  - Design based on Hidden service in IP [6]
    - To prevent information leakage from content name and signature
      - Producers advertise self-certifying names as their **pseudonyms** and communicate with consumers **through rendezvous points without using their routable names**
      - Producers use self-signed certificates
    - To prevent information leakage from packet route
      - Producers communicate with other nodes only through circuits
  - Leverage anonymity offered in network by using RICE [7]
    - Leverage the feature of NDN that anonymity of a sender of Interest packets is naturally achieved

System Overview

1. A producer generates her/his pseudonym called an **onion name**
2. The producer asks an AR to act as a **rendezvous point**
3. The producer uploads her/his **descriptor** to several ARs called **descriptor directories**
   - The descriptor contains information about which rendezvous point to use when a consumer wishes to retrieve content of a certain onion name
4. A consumer who learns the onion name in some out-of-band way downloads the descriptor
5. The consumer issues content requests specifying the onion name through the rendezvous point
   - Because the rendezvous point just forwards Interest packets along a circuit built by the producer, it does not learn the producer's identity



Protocol #1 Naming

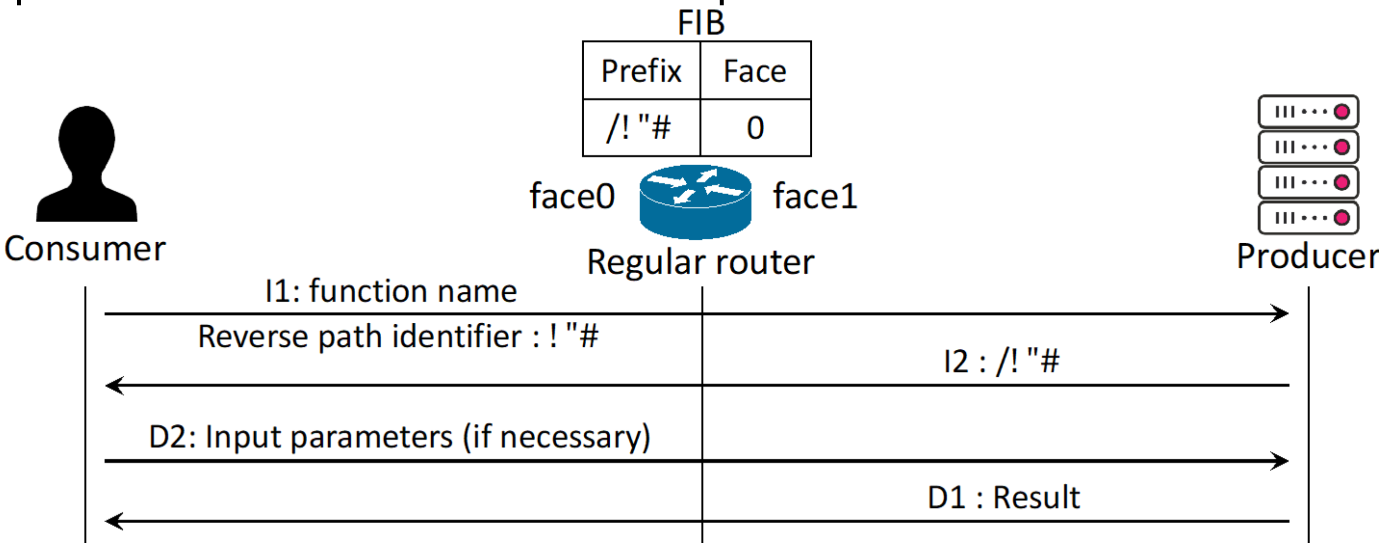
- Onion name structure : **/onion/$\boldsymbol{Hash(pk_{id})}$/⟨suffix⟩**
  - (pk_{id}, sk_{id}) is a public/private keypair of a producer
- Features
  - Onion name does not reveal information on the producer because it is
    - non routable
      - If routable, adversaries can identify a producer by sending an Interest packet directly to the producer and tracking it
    - non human-readable
      - If human-readable, the producer's information can be revealed
  - Onion name is secure because it is
    - unique and self-certifying
      - If not, a consumer cannot confirm whether the origin of received content is the intended producer

Protocol #2 Rendezvous Point Establishment

- Goal
  - The producer asks an AR to act as a rendezvous point by sending the onion name and a self-signed certificate Cert(pk_{id})
  - The AR accepts it if $Hash(pk_{id})$ contained in the onion name is valid for pk_{id} in the certificate
- Problem
  - The producer cannot send these elements with the standard Interest-Data exchange
    - This is because the producer's routable name must be hidden to all other entities to achieve producer anonymity
- Solution
  - By leverage 4-way handshake in RICE, the producer enables an AR to send back Interest packets along reverse paths without advertising the routable name

RICE Overview

- Original goal
  - To enable consumers to delegate computation to remote entities
- Procedure
  - A consumer issues an I1 packet carrying a function name
    - I1 packet also carries a consumer-chosen **reverse path identifier** : rID
  - Each intermediate router creates an ephemeral FIB entry pointing to the face from which the I1 packet came
    - The sequence of FIB entries is called a **reverse path**
  - A producer sends back I2 packet(s) along the reverse path to let the consumer return some input parameters for the function with the corresponding D2 packet(s)
  - The producer returns the result in a D1 packet or in another Interest-Data exchange



FIB

Prefix	Face
/! "#	0

Protocol #2 Rendezvous Point Establishment

- Procedure
  - The producer builds a circuit consisting of an AR that is a candidate for a rendezvous point (/RP) and another AR (/AR)
  - The producer sends the onion name and a self-signed certificate Cert(pk_{id}) by using RICE-based 4-way handshake
  - If Cert(pk_{id}) and the onion name is valid, the AR (/RP) starts to act as a rendezvous point

Notations

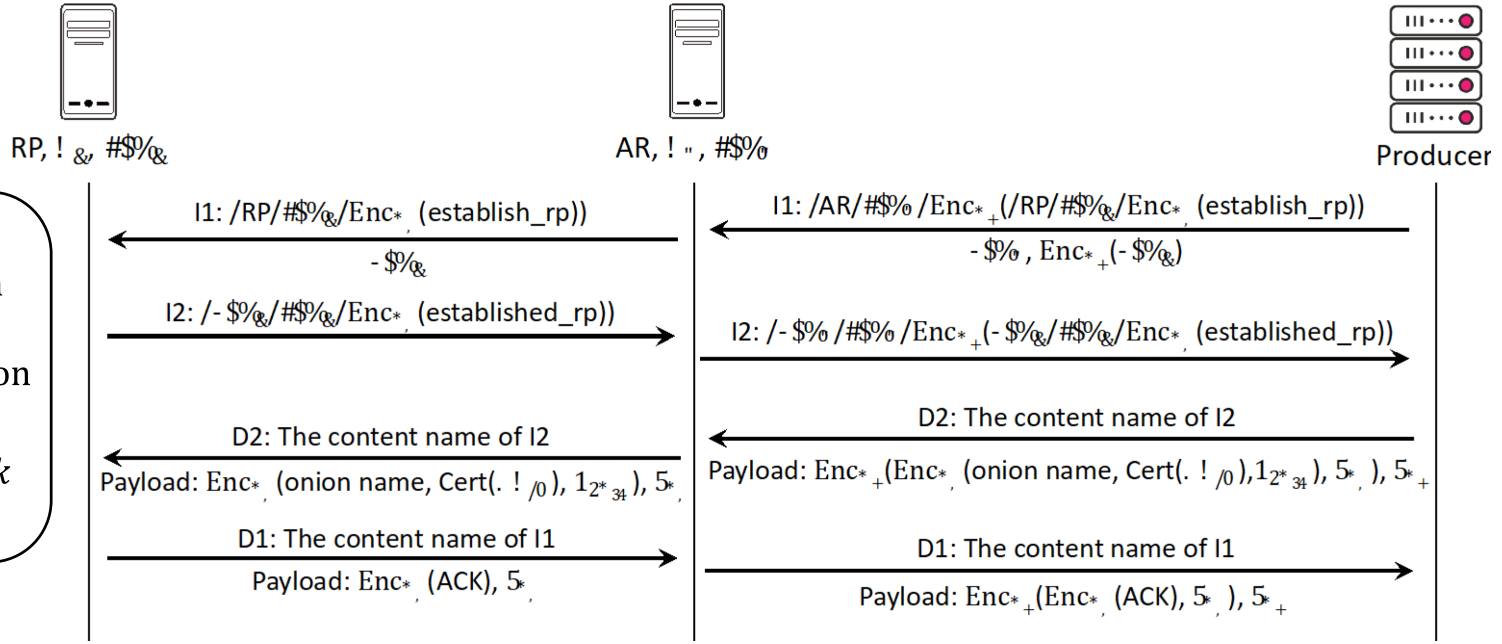
sID_i : Identifier indicating which secret key is used

Enc/Dec : Encryption/Decryption

rID_i : Reverse path identifier

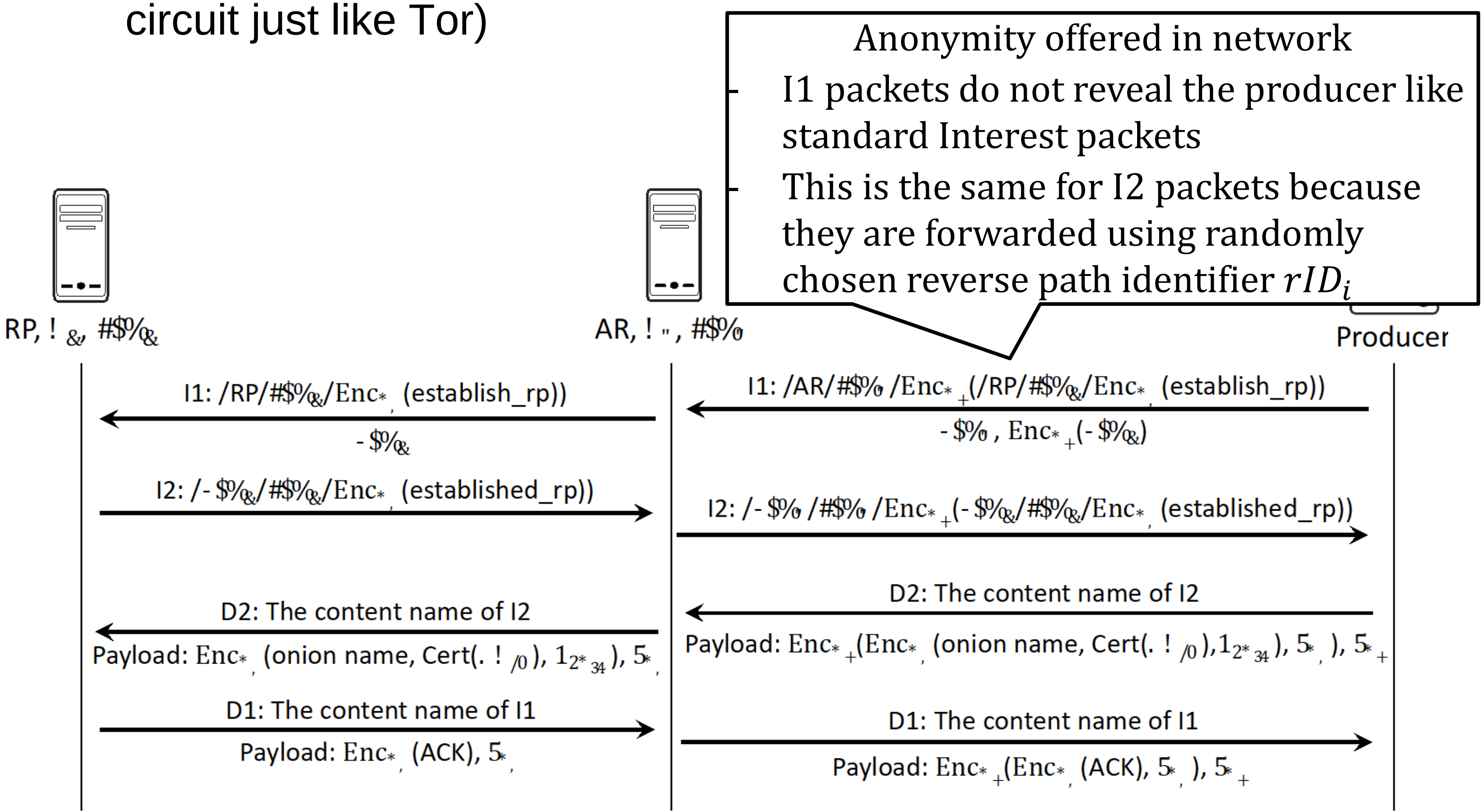
σ_{sk} : signature generated with sk

t_k : MAC tag generated with k


RP, ! &, #$%&
AR, ! ", #$%
Producer
I1: /RP/#$%&/Enc* , (establish_rp))
- $%&
I2: /- $%&/#$%&/Enc* , (established_rp))
D2: The content name of I2
Payload: Enc* , (onion name, Cert(. ! /0), 12* 34), 5* ,
D1: The content name of I1
Payload: Enc* , (ACK), 5* ,
I1: /AR/#$% /Enc* + (/RP/#$%&/Enc* , (establish_rp))
- $% , Enc* + (- $%&)
I2: /- $% /#$% /Enc* + (- $%&/#$%&/Enc* , (established_rp))
D2: The content name of I2
Payload: Enc* + (Enc* , (onion name, Cert(. ! /0),12* 34), 5* ,), 5* +
D1: The content name of I1
Payload: Enc* + (Enc* , (ACK), 5* ,), 5* +

Our System vs Hidden service in IP

- Advantage of our system
  - Like ANDaNA, our system achieves a level of anonymity comparable to hidden service with one fewer ARs thanks to anonymity offered in network
    - (Hidden service use three ARs (including a rendezvous point) in each circuit just like Tor)



Anonymity offered in network

- I1 packets do not reveal the producer like standard Interest packets
- This is the same for I2 packets because they are forwarded using randomly chosen reverse path identifier rID_i

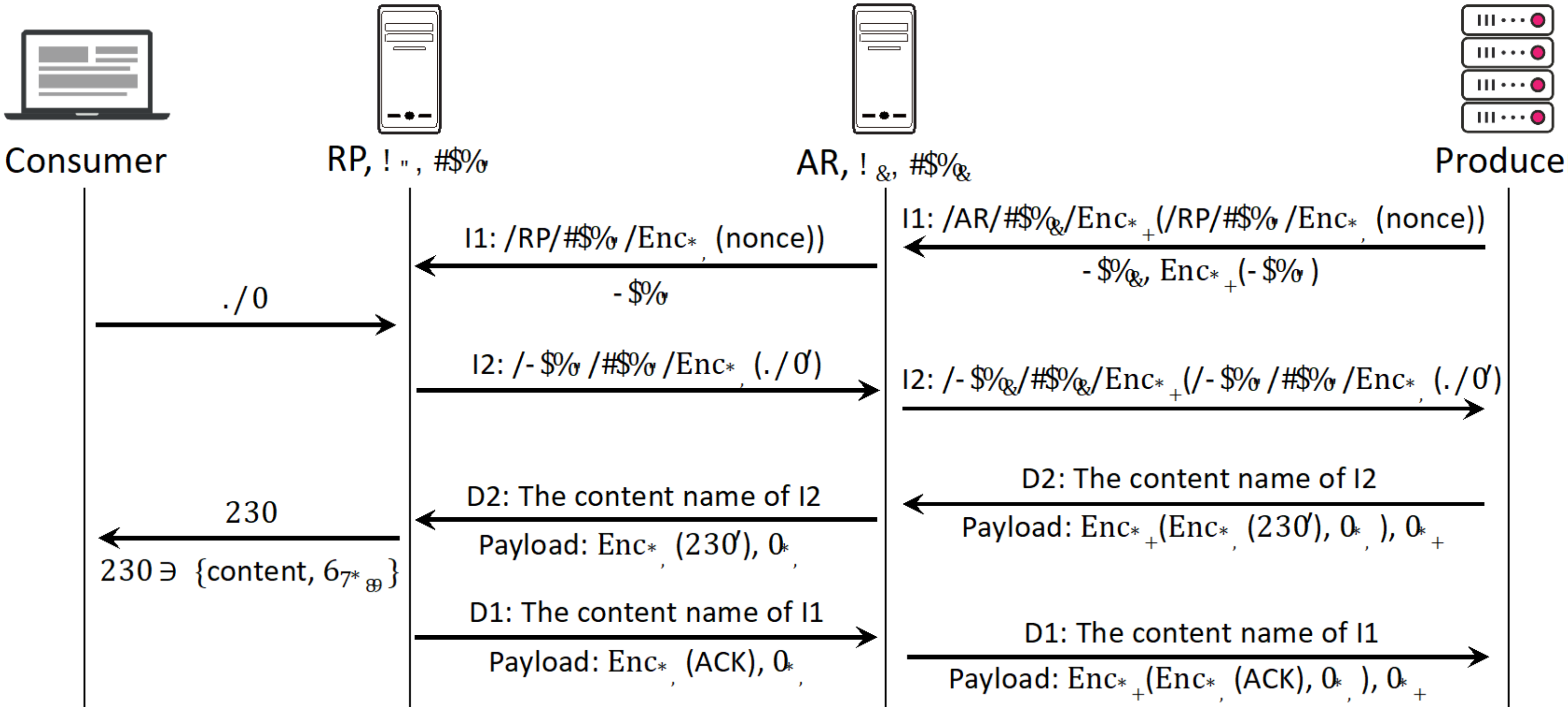
RP, ! &, #$%&
AR, ! ", #$%0
Producer
I1: /RP/#$%&/Enc*, (establish_rp))
- $%&
I2: /- $%&/#$%&/Enc*, (established_rp))
D2: The content name of I2
Payload: Enc*, (onion name, Cert(. ! /0), 12* 34), 5*,
D1: The content name of I1
Payload: Enc*, (ACK), 5*,
I1: /AR/#$%0 /Enc*+(/RP/#$%&/Enc*, (establish_rp))
- $%0 , Enc*+(- $%&)
I2: /- $%0 /#$%0 /Enc*+(- $%&/#$%&/Enc*, (established_rp))
D2: The content name of I2
Payload: Enc*+(Enc*, (onion name, Cert(. ! /0),12* 34), 5*,), 5*+
D1: The content name of I1
Payload: Enc*+(Enc*, (ACK), 5*,), 5*+

Protocol #3,4 Descriptor Publication/Retrieval

- Goal
  - The producer uploads the descriptor to several descriptor directories to enable consumers to find the established rendezvous point corresponding to the onion name
- Descriptor
  - Name : /onion/$Hash(pk_{id})$/descriptor
  - Payload : $\mathrm{Cert}(pk_{id})$ and the routable name of the rendezvous point
  - Signed with sk_{id}
  - Descriptor directories are ARs managed based on a distributed hash table (DHT) and the responsible directories are determined by the descriptor name and current timestamp
- Procedure
  1. The producer upload the descriptor by using the 4-way handshake
  2. A consumer derives the descriptor name from $Hash(pk_{id})$ contained in an onion name
  3. The consumer finds the responsible descriptor directories and downloads the descriptor

Protocol #5 Content Publication

- Procedure
  - Content publication phase also use the 4-way handshake in RICE
  - The producer continuously creates reverse paths between the rendezvous point
  - On the receipt of an Interest packet (*int*) from a consumer, the rendezvous point forwards it as an I2 packet (*int*’) in a reverse path
  - The corresponding Data packet (*dat*) is returned as a D2 packet (*dat*’) in the reverse path



Consumer
RP, ! ", #$%
AR, ! &, #$%&
Producer
I1: /RP/#$% /Enc* , (nonce))
- $%
I1: /AR/#$%&/Enc* +(/RP/#$% /Enc* , (nonce))
- $%&, Enc* +(- $%)
./0
I2: /- $% /#$% /Enc* , (./0′)
I2: /- $%&/#$%&/Enc* +(/- $% /#$% /Enc* , (./0′)
230
230 ∋ {content, 67* 89}
D2: The content name of I2
Payload: Enc* , (230′), 0* ,
D2: The content name of I2
Payload: Enc* +(Enc* , (230′), 0* ,), 0* +
D1: The content name of I1
Payload: Enc* , (ACK), 0* ,
D1: The content name of I1
Payload: Enc* +(Enc* , (ACK), 0* ,), 0* +

Performance Evaluation

- Implementation
  - We implemented our system as applications that run on producers and ARs by using the ndn-cxx library
    - Encryption/decryption algorithm : AES-128
    - MAC generation/verification algorithm : HMAC with SHA-256
- Performance in content publication
  - Assume a simple line topology
  - Fig. 1 and 2 shows the overall **process delay** and the **throughput** of the applications as a function of the achieved level of anonymity, respectively
    - Level of anonymity = n means the anonymity offered when n ARs are used in hidden service
    - **Our system has better performance** because our system reduces the number of required ARs, and thus the number of cryptographic operations, in a circuit by one while still achieving the same level of anonymity

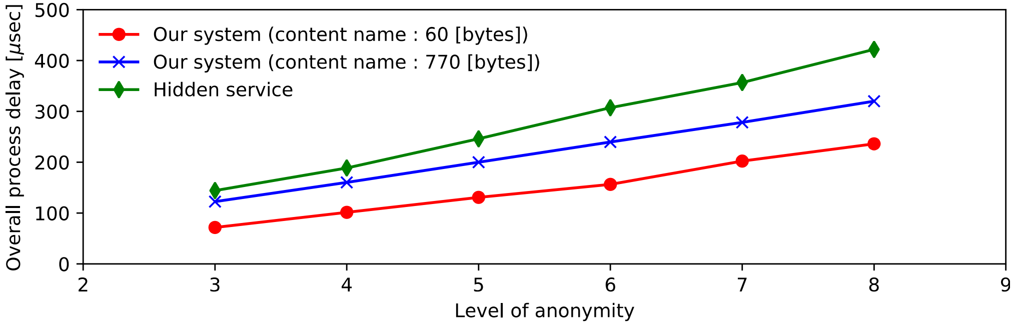


Fig. 1

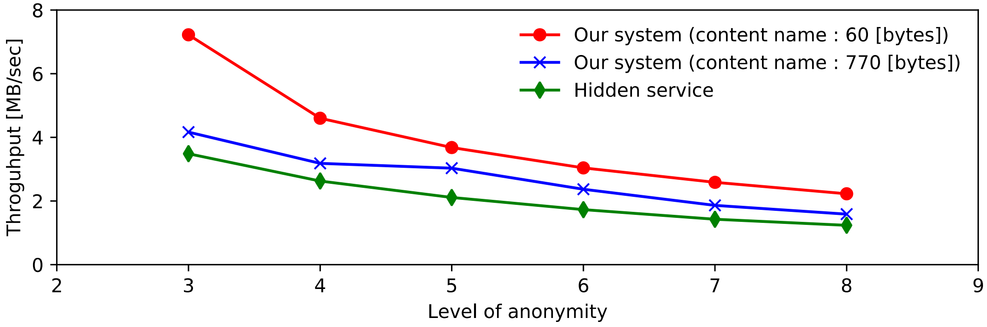
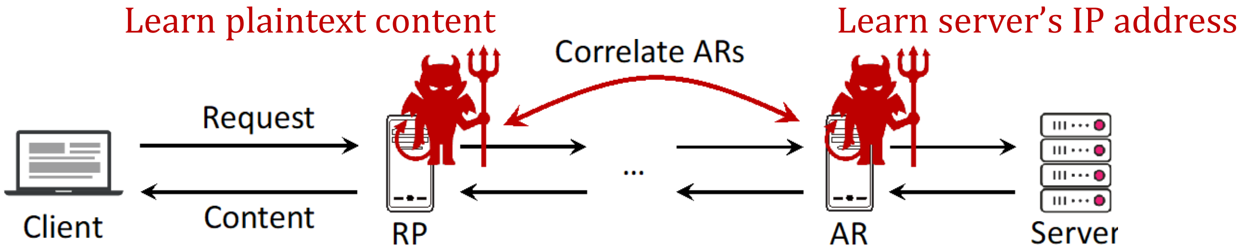


Fig. 2

The Predecessor Attack

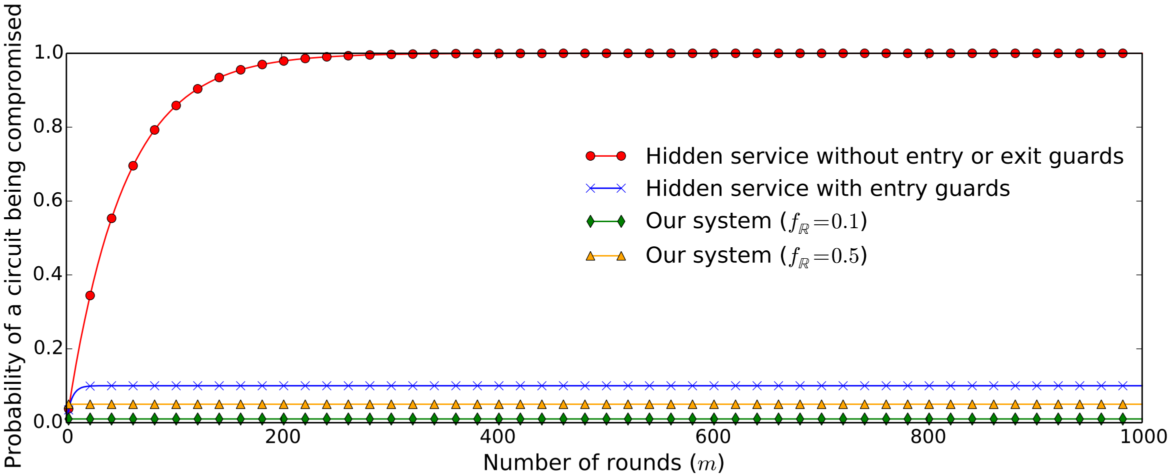
- Predecessor attack in IP [8]
  - Assumption and notation
    - Assume that an adversary can always correlate two entities included in the same circuit by **traffic analysis** using timing and volume of packets if both of them are compromised
    - Then, the adversary breaks anonymity if both the first and the last AR are compromised
      - Because such an adversary can correlate plaintext packet and the server



    - **Round** : a series of communication performed without changing a circuit
  - Attack
    - If all the ARs in circuits are chosen uniformly at random in each round, the probability that anonymity is broken grows to 1.0 as the number of rounds increases
      - i.e.) Anonymity will definitely be broken in the future
      - This is because compromised ARs will eventually be chosen as the first and the last AR in circuit

Success Probability of Predecessor Attack

- Hidden service : f_A (< 1.0)
  - f_A : The probability that each AR is compromised
  - Use **entry guard** : the first-hop AR which is repeatedly used for circuits
    - The adversary must compromise the entry guard to break anonymity
- Our system : $f_R \times f_A$ $(< f_A)$
  - f_R : The probability that each (layer 3) router is compromised
  - Use **entry guard and exit guard**
    - The first-hop router of a producer plays the role of the entry guard
    - In addition, **the last-hop AR, called exit guard, is fixed**
    - The adversary must compromise both the entry guard and the exit guard to break anonymity



Future Work

- Conduct more performance evaluations under various scenarios, e.g., mobile wireless networks and congested networks

- Implementing the protocol on Cefore, which is provided by NICT

- Integrating several DoS mitigation mechanisms into our system
  - E.g.) requiring producers to solve puzzles, which cost a lot of CPU cycles or memory before establishing reverse paths and circuits, can hinder adversaries from making routers and ARs unavailable by establishing many reverse paths and circuits through them

Acknowledgement

- The research leading to these results has received funding from NICT (Contract No. 191).

References

[1] S. DiBenedetto et al., “ANDaNA: Anonymous named data networking application,” ArXiv e-prints, Dec. 2011.

[2] P.Zhang et al., “Achieving content-oriented anonymity with crisp,” in IEEE Transactions on Dependable and Secure Computing, vol. 14, no. 6, pp. 578–590, Nov 2017.

[3] S. Kaushik et al., “Ndn- abs: Attribute-based signature scheme for named data networking,” in Proceedings of the ACM Conference on ICN, Ser. ACM-ICN ’19, New York, NY, USA, 2019, pp. 123–133.

[4] A. Ghodsi et al., "Naming in content-oriented architectures." in Proceedings of the ACM SIGCOMM workshop on ICN. 2011.

[5] Z. Zhang et al., "An Overview of Security Support in Named Data Networking," in IEEE Communications Magazine, vol. 56, no. 11, pp. 62-68, November 2018.

[6] The Tor Project. (2017) Tor rendezvous specification - version 3. [Online]. Available:http://gitweb.torproject.org/torspec.git/tree/rend-spec-v3.txt

[7] M. Krol et al., “Rice: Remote method invocation in icn,” in Proceedings of ACM Conference on ICN, 2018.

[8] L. Overlier et al., “Locating hidden servers,” in 2006 IEEE Symposium on Security and Privacy (S P’06), May 2006, pp. 15 pp.–114.