Virtual & Invisible Private Network a Zero-Trust Architecture for Anonymous Communication on the Internet (WiP)

Frédéric Laurent[1], **Baptiste Polvé**[1], **Guillaume Nibert**[1,2], Alexis Olivereau[3]

[1]Snowpack, F-91400 Orsay, France
[2]Sorbonne Université, CNRS, LIP6, F-75005 Paris, France
[3]Université Paris-Saclay, CEA, List, F-91120, Palaiseau, France

32nd Computer & Electronics Security Application Rendezvous (C&ESAR)
Couvent des Jacobins, Rennes, France, 19-20 November 2025

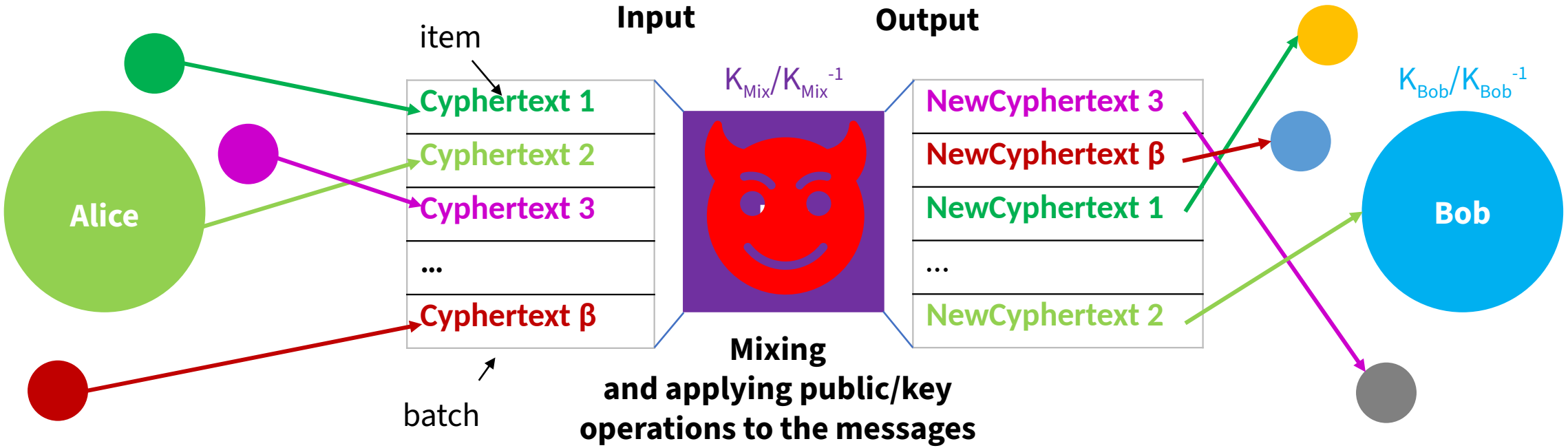
baptiste.polve@snowpack.eu
guillaume.nibert@snowpack.eu

Outline

Related work
Motivation and Goals
Virtual & Invisible Private Network Design and Protocol
Evaluation
Early experiences: VIPN in the Wild
Work in Progress
References
Appendices

Related work (1/5)

[1] D. L. Chaum, *Untraceable electronic mail, return addresses, and digital pseudonyms*, Commun. ACM, vol. 24, no. 2, pp. 84–90, Feb. 1981, doi: 10.1145/358549.358563.



M: a message.
K_A : public key of A.
K_A^{-1} : private key of A.
$K_A^{-1}(K_A(M)) = (K_A(K_A^{-1}(M)) = M$

Anonymity lost
(Trusted Tier Party)

High Latency

Asymmetric cryptographic operations **only**.

Usage : End-to-End

Related work (2/5)

Mixnets

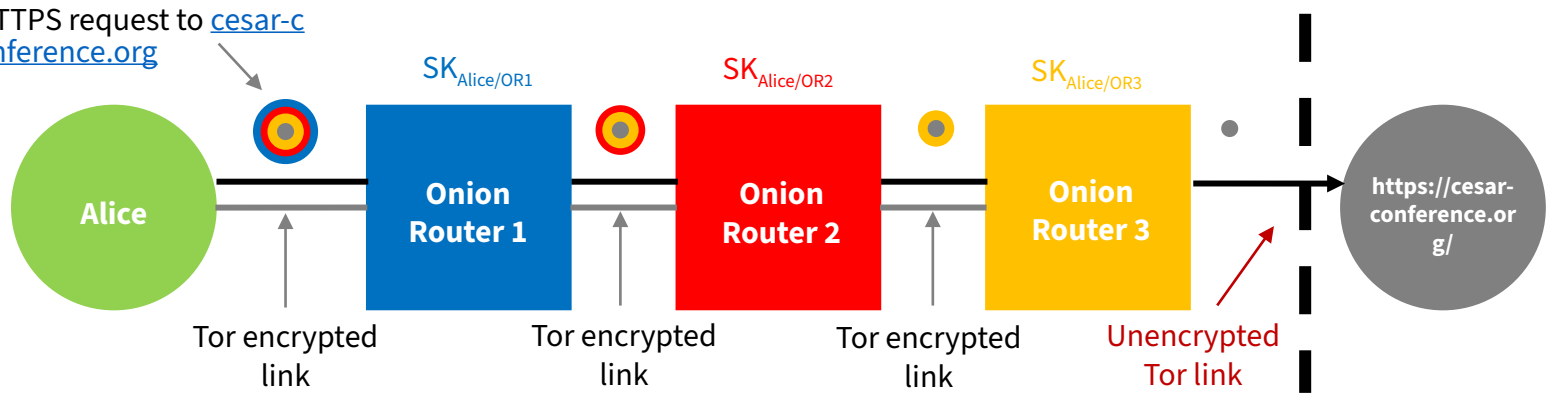
Onion routing and **Tor-based protocols**

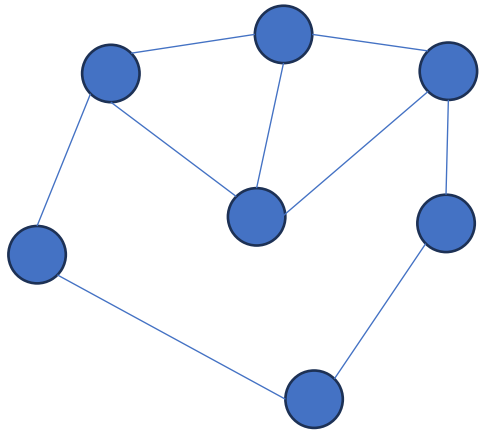
Random Walks and **DHT-Based protocols**

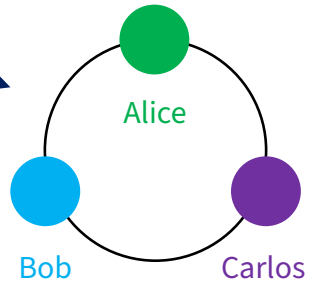
DC-Nets

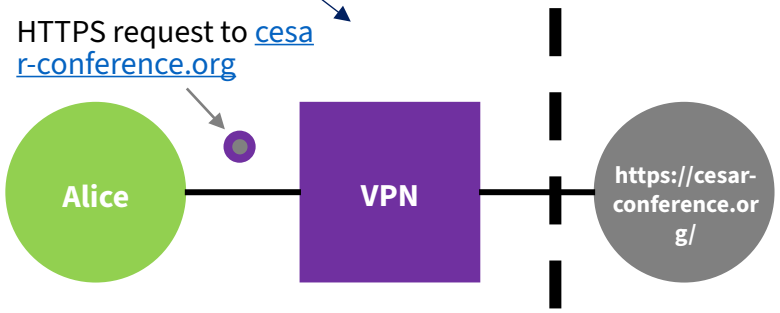
Unclassified

(VPNs)









Related work (3/5)

Reference	Year	Name	Class	Summary	Limitations
Van den Hoof et al. [3]	2015	Vuvuzela	Mixnet	- dialing protocol with dread drops + conversation protocol - expose less metadata : total number of users in a conversation + total number not in a conversation - cover trafic (noise) over time + differential privacy used for proving privacy quarantees.	Application: limited to a private chat, 5 million users max, operate in synchronous round →real time apps not working.
Tyagi et al. [4]	2017	Stadium	Mixnet	Horizontal scaling improvement of Vuvuzela. Multiple chains, treating specific messages. Verifiable parallel mixnet. Differential privacy + cover trafic. 4x more users. 7-15x lower bandwidth cost.	High latency →not usable with real time apps (100 users, 6.3 Musers, chain length 3, bw 18 Mb/s, 44 seconds) – up to 876 s
Gelernter et al. [5]	2018	AnonPoP	Mixnet	Similar to Vuvuzela : covert raffic + dead drops = post office servers + scalable + Support device disconnection + tagging attack detection and bad-server isolation, 2 cents/month per client.	Limited to email and messaging + Some active analysis attacks are still feasables (overload of honest server, bad server iso not fully analysed…)
Kwon et al. [6]	2017	Atom	Mixnet	Overcomes secure low-latency communications challenge by scaling horizontally. Servers grouped into "anytrust" groups. 1 Honest. No fixed cascade → variant of ElGamal for encryption/decryption + Fault tolerance.	Application : limited to microblogging and dialing. Synchronous round, High Latency 28min, 1 Musers →real time apps not working.
Kwon et al. [7]	2015	Riffle	Mixnet	Combination of verifiable shuffling + onion encryption with symmetric-key private information retrieval (PIR). 1 honest server is enough in the chain.	Application : limited to filesharing (BW: 100 KB/s, 200 users) and microblogging (latency: 10s, 100000 users). Group communication.

Related work (4/5)

Reference	Year	Name	Class	Summary	Limitations
Chaum et al. [8]	2016	cMix	Mixnet	Reducing latency and computational cost on user devices -> precomputation before the communication by servers, not end-to-end devices.	Application : chats (xxMessenger), but not real-time applications. Does not scale horizontally.
Piotrowska et al. [9]	2017	Loopix	Mixnet	Poisson Mixing, Stratified Topology, Asynchrony, Cover trafic, n-1 & replay active attack resistance, better latency than Vuvuzela and similar.	Application : Up to IP with NYM, but even with lower latency still not able to do real-time applications.
Dingledine et al. [11]	2004	Tor	Onion Routing (Goldschlag et al., 1996 [10])	Data is encrypted multiple times like an onion. Like a mixnet cascade without shuffle. 2 modes browsing and E2E	Vulnerable to various traffic analysis attacks : e.g. time correlation + Dos, etc.; Application : Only TCP
Clarke et al. [12]	2001	Freenet (Hyphanet)	Random walks & DHT	Peer-to-peer, decentralised file-sharing and storage system. Indirection. Hops-to-live. Replication and Plausible deniability	Application : file-sharing and storage only. Active attacks, quite rare : key collision...
Bennett and Grothoff [13]	2004	GNUnet (GAP)	Random walks & DHT	P2P network. Credit-based routing system, relaying = earn credits; sending = use credit. Direct comm between sender an recipient is possible. Goal: decorrelate sender and its actions (≠ Indirection). Threshold mixes. Time to live.	File sharing. Not adapted for real-time apps.
Wang and Borisov [14]	2012	Octopus	Random walks & DHT	Anonymous DHT: prevent an adversary from determining who is searching for what. Active attack resistance. Detect and remove malicious peers. Check if neighbour or specific node is manipulating successor tables (surveillance) Conceals real queries by splitting them across	Application : search and access files. Not adapted for real-time apps. False alarm when node are not stable →removal of honest nodes

Related work (5/5)

Reference	Year	Name	Class	Summary	Limitations
Corrigan-Gibbs and Ford [15]	2010	Dissent	DCnet/Mixnet	Provide anonymity within small groups. Send messages to each other, or whole group, or non-member. Integrity and Accountability. Multiparty computation and layered encryption. Shuffling and bulk transfer.	Not designed for large-scale communication. Synchronous rounds →Not usable for realtime apps.
Zantout and Haraty [16]	2001	I2P	Unclassified	Garlic Routing, end-to-end communication + DHT	Slow, not good for low latency applications.
Sherwood et al. [17]	2005	P5	Unclassified	Tree-like structure to broadcast messages. Groups. Trade-off between communication efficiency and the level of anonymity	Not scalable. High overhead if groups are composed of a lot of users. Not usable with high bandwidth.
Shokri et al. [18]	2007	CAR	Unclassified	Nodes along a path are linked in a virtual chain, unicast-based broadcasting for route discovery. Wireless ad hoc networks.	Ad-hoc; useful locally, not for exchanging messages over the globe.

Outline

Motivation and Goals
Threat actors and adversaries

Threat actors	Adversaries		
Individual hacker ISP Data center operator Cloud provider	Local	Polynomially-bounded	Active / Passive	Internal / External
Core network operator Nation-State	Global	Polynomially-unbounded	Active / Passive	Internal / External

Has a partial view of the network, controls some links.

E.g. cannot break TLS/RSA/AES, computationally bounded schemes

Has a complete view of the network, controls all the links.

E.g. can break TLS/RSA/AES… ;cannot break information-theoric schemes (One-Time Pad)

Anonymity and Security Goals

Sender anonymity [19]

Sender-Receiver unlinkability [19]

Secrecy

Transparent man-in-the-middle resistance

Service Goals

Low latency and high-capacity

Scalability

Compatibility

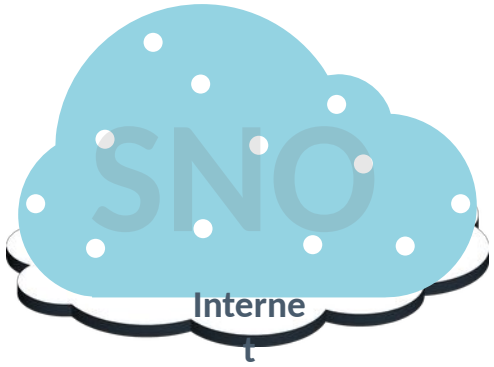
(Usability)

Outline

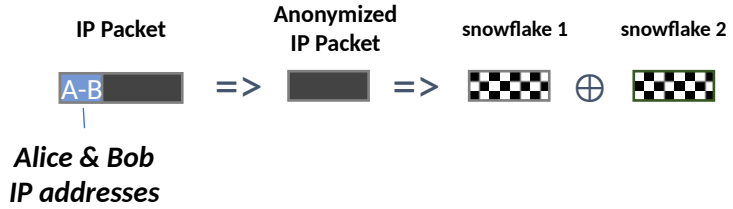
Related work
Motivation and Goals
Virtual & Invisible Private Network Design and Protocol
Evaluation
Early experiences: VIPN in the Wild
Work in Progress
References
Appendices

Virtual & Invisible Private Network Design and Protocol Principles

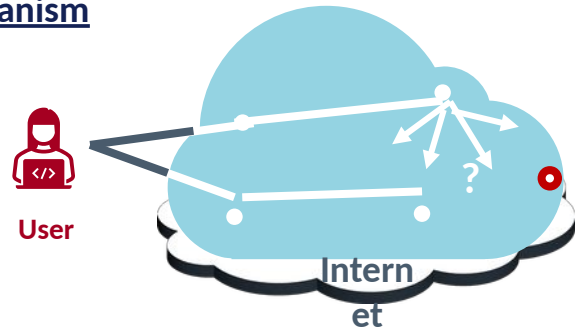
1. Distributed network overlay (SNO)



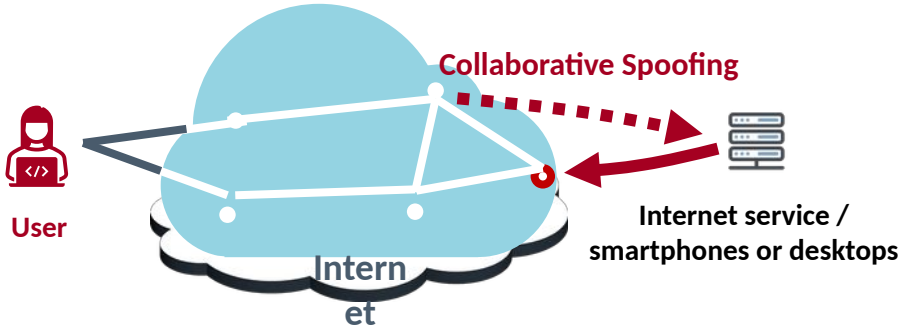
2. Packets fragmentation into anonymized noise (snowflakes) via OTP



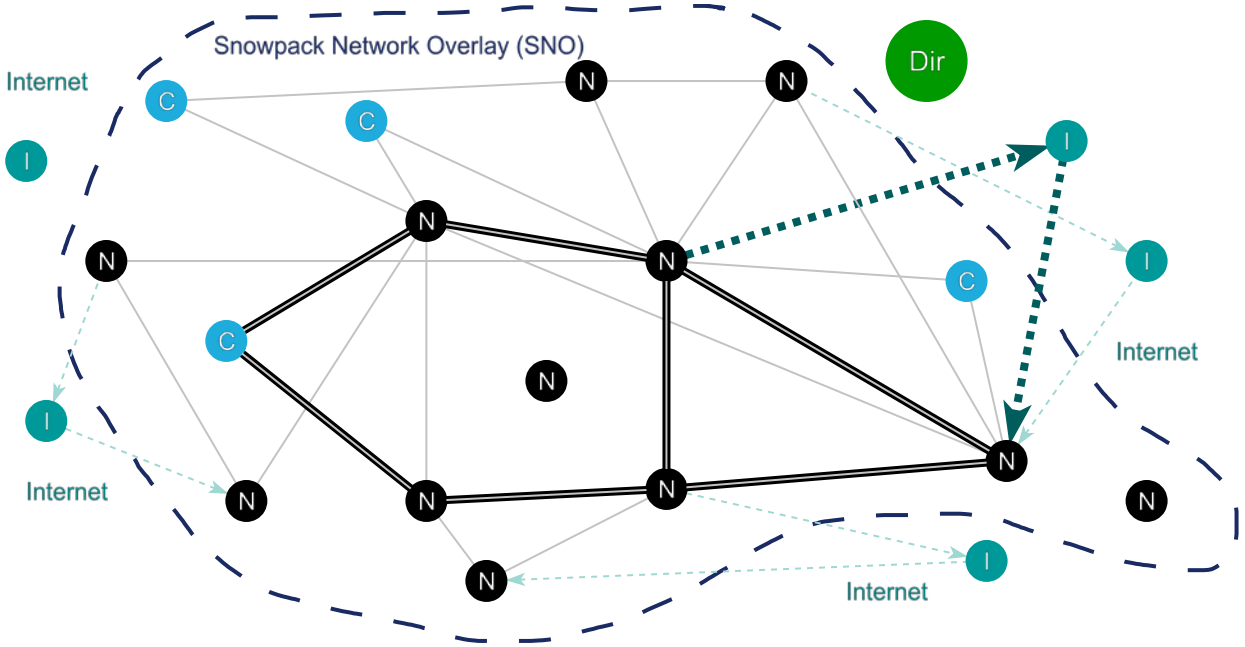
3 + 4. Split of information into routes, nodes autodiscovery mechanism

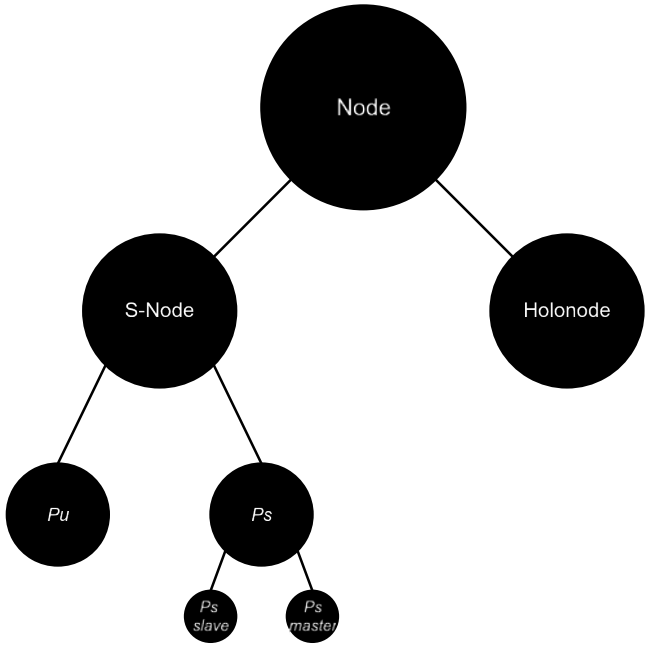


5. Distributed exit nodes



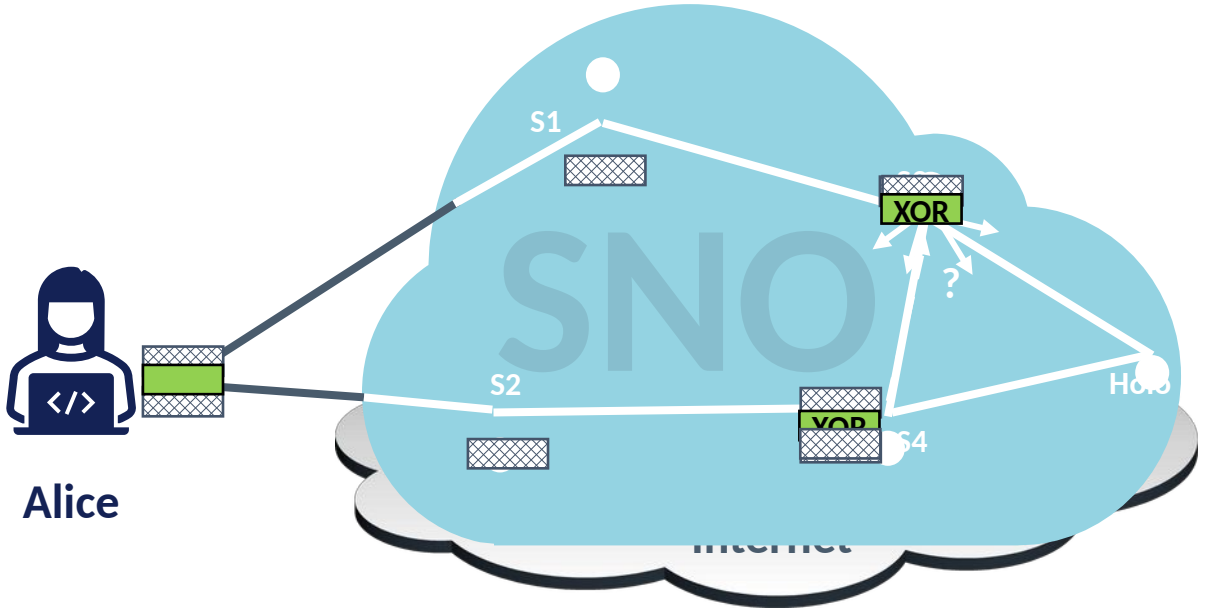
Virtual & Invisible Private Network Design and Protocol Components





Existing roles in a VIPN route

Virtual & Invisible Private Network Design and Protocol
Route Construction



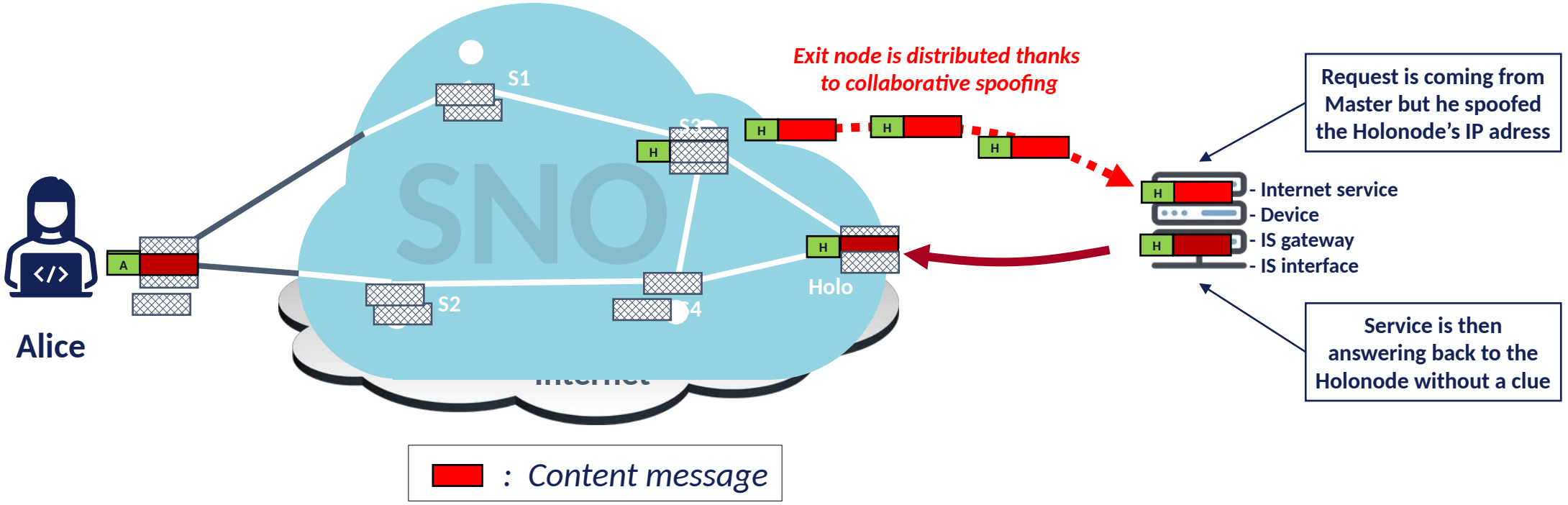
: *Autodiscovery message*

SECRET = Token | hash (Token | IP_addr_{S3} | IP_addr_{S4}) | $IP\_addr_{Holonode}$

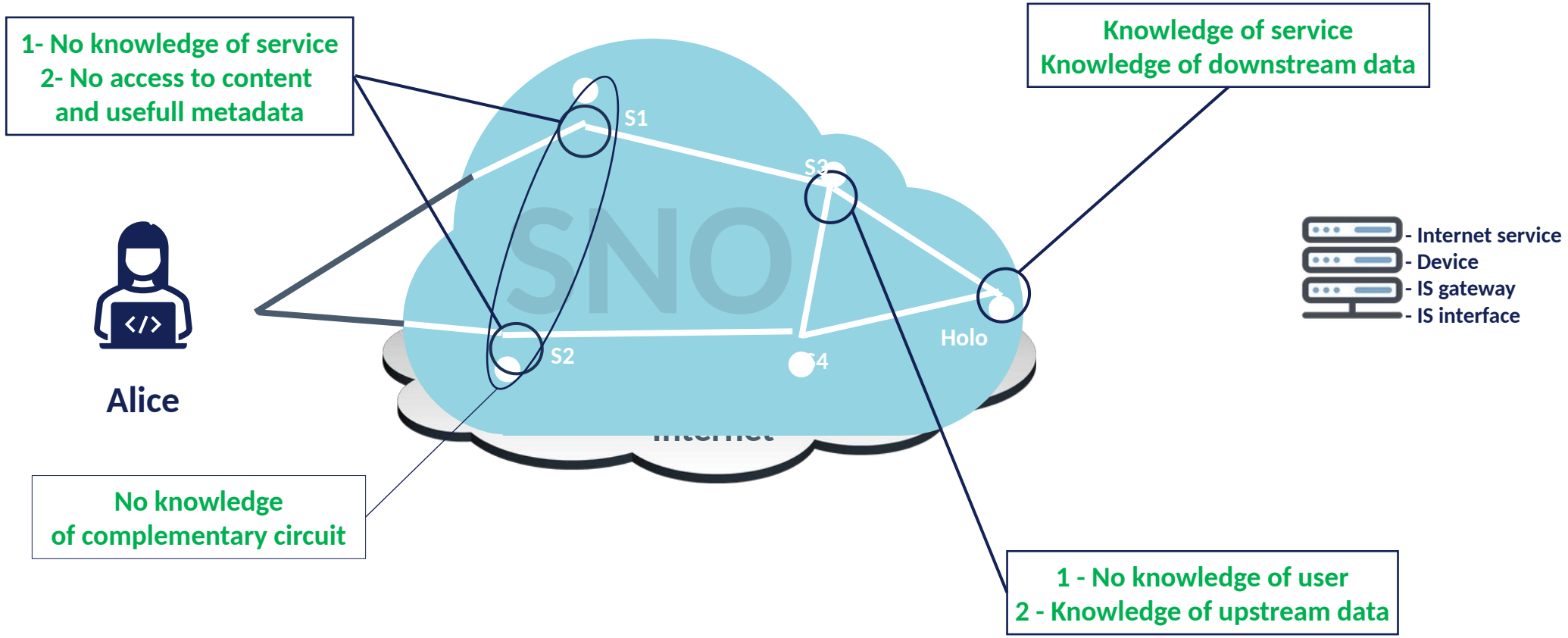
: *Autodiscovery Snowflake_S3*

: *Autodiscovery_Snowflake_S4* ⊕ SECRET = *Autodiscovery_Snowflake_S3*

Virtual & Invisible Private Network Design and Protocol
Nominal mode



Virtual & Invisible Private Network Design and Protocol
Observed properties

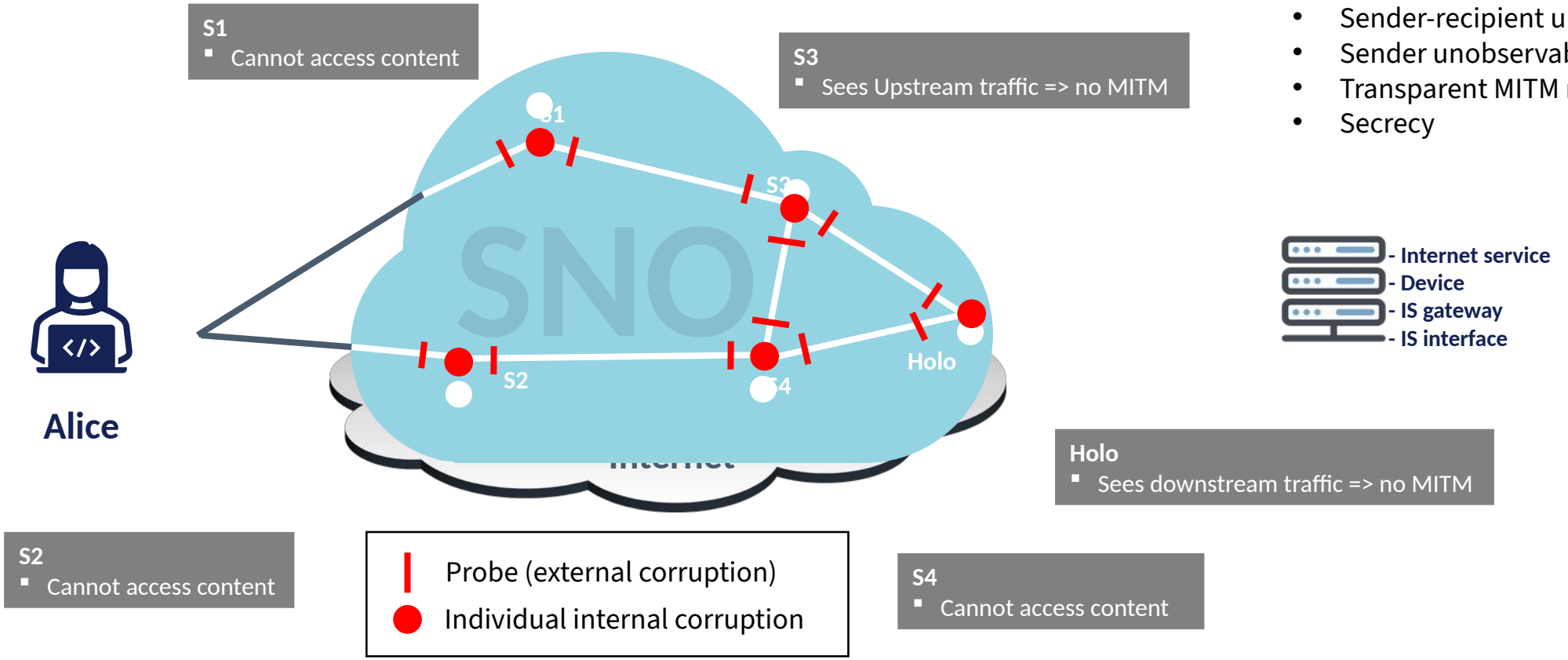


Outline

Related work
Motivation and Goals
Virtual & Invisible Private Network Design and Protocol
Evaluation
Early experiences: VIPN in the Wild
Work in Progress
References
Appendices

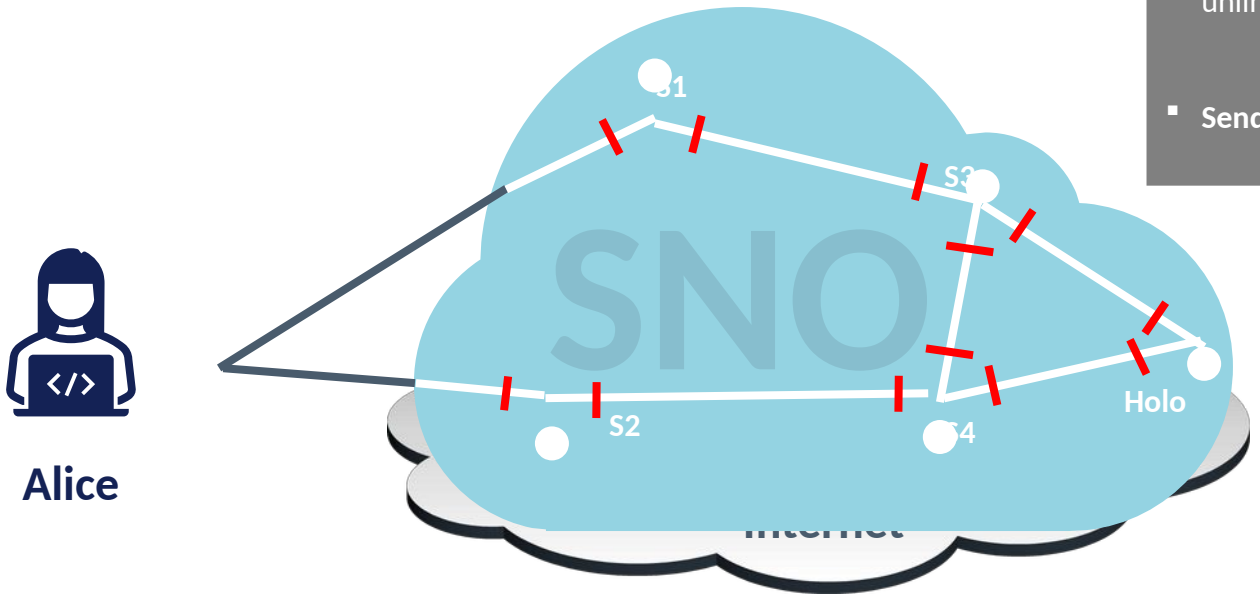
Local polynomially bounded and unbounded adversary (external and internal)

- Sender anonymity
- Sender-recipient unlinkability
- Sender unobservability
- Transparent MITM resistant
- Secrecy



Global active polynomially *bounded* adversary

GAA *bounded*
- Cannot decrypt TLS => **Secrecy**
- Cannot actively modify the traffic => **Transparent MITM resistance**
- Can add delays to break sender anonymity and sender-recipient unlinkability, but complex to add.
  - Cannot watermark snowflakes (TLS 1.3) -> **partial guarantee of sender-anonymity and sender-recipient unlinkability**
- **Sender unobservability is not offered by default.**

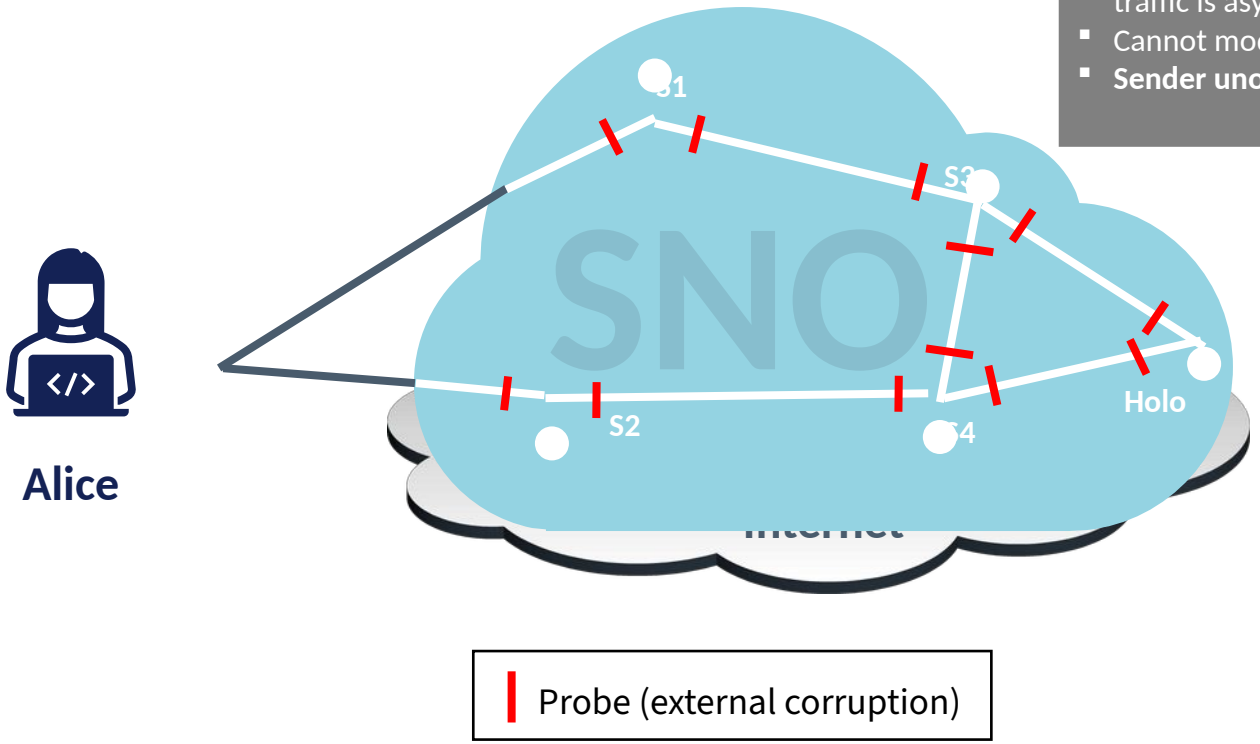


- Internet service
- Device
- IS gateway
- IS interface

Probe (external corruption)

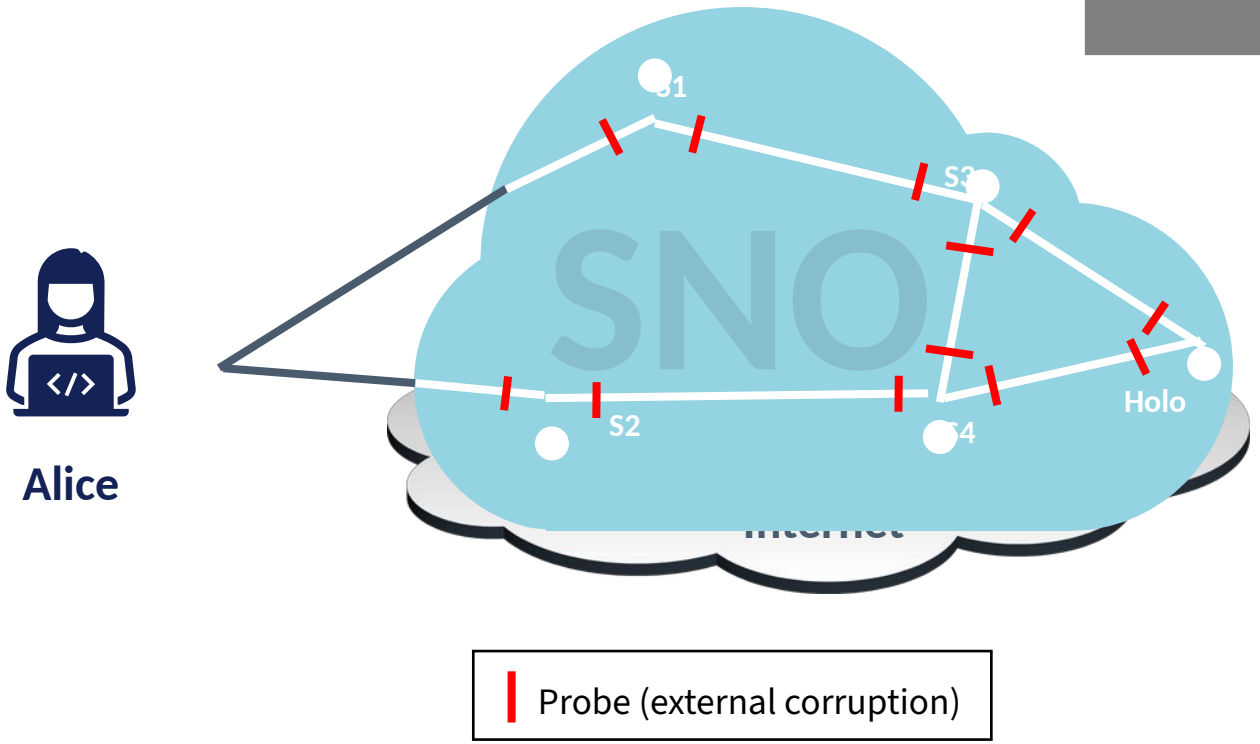
Global passive polynomially *bounded* adversary

GPA *bounded*
- Cannot decrypt TLS => **Secrecy**
- Cannot follow the traffic => **Sender anonymity**
- Cannot distinguish users from monitoring inbound and outbound as the traffic is asymmetrized => **Sender-recipient unlinkability**
- Cannot modify the traffic (because passive) => **Transparent MITM resistance**
- **Sender unobservability is not offered by default.**



Global active polynomially *unbounded* adversary

GAA *unbounded*, no guarantee at all:
- Can decrypt TLS and recombine every snowflakes => **No secrecy**
- Can so determine number of circuits => **No anonymity at all.**
- Can manipulate the traffic to break **transparent MITM resistance.**



- Internet service
- Device
- IS gateway
- IS interface

Probe (external corruption)

Evaluation
Summary

Adversaries				Properties				
				Sender anonymity	Sender-recipient unlikability	Sender unobservability	Secrecy	Transparent MITM resistant
Passive adversary	GPA			X	X	X*	X	X
Active adversary	Corruption power							
	Polynomially bounded	Local	Pus	X	X	X	X	X
			Master	X	X	X	X	X
			Slave	X	X	X	X	X
			Holonode	X	X	X	X	X
		Global		~	~	X*	X	X
	Polynomially Unbounded	Local	Pus	X	X	X	X	X
			Master	X	X	X	X	X
			Slave	X	X	X	X	X
			Holonode	X	X	X	X	X
		Global		.	.	.	.	.

X means that the property is verified.
X* means that the property is not offered by default but is verified if the user sends useless packets.
~ means that the property is partially verified: it is verified on the basis of an uncalculated minimum quantity of network traffic processed by the adversary.

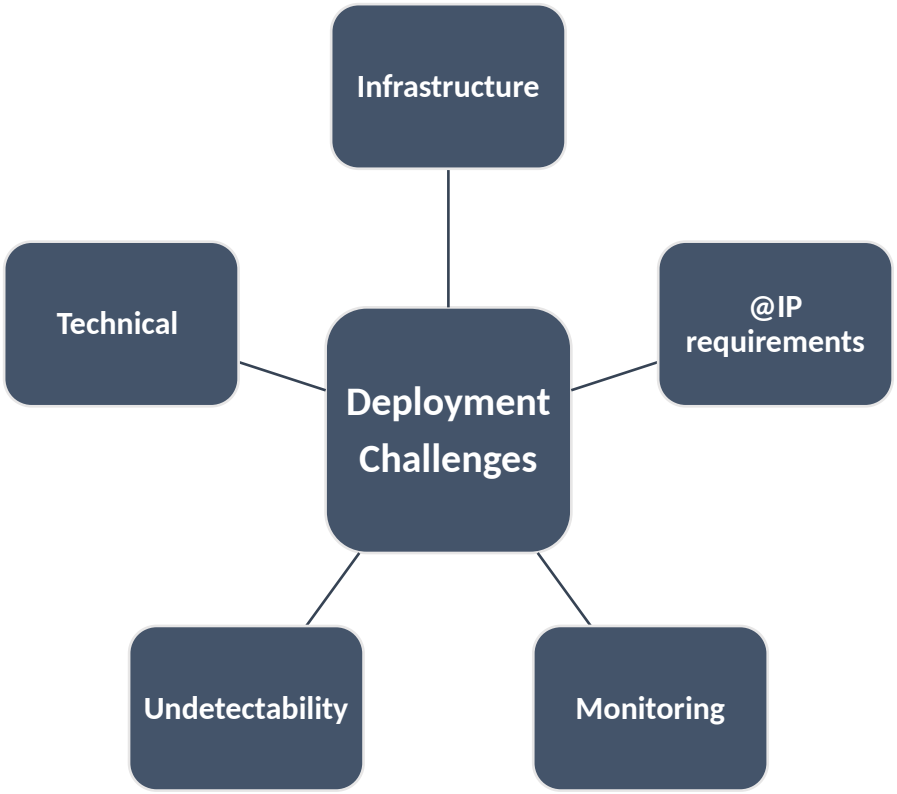
Outline

Related work
Motivation and Goals
Virtual & Invisible Private Network Design and Protocol
Evaluation
Early experiences: VIPN in the Wild
Work in Progress
References
Appendices

Evaluation
Early experiences: VIPN in the Wild

Live since 2022

Bandwidth **20-150 Mbps**	Latency **+ 40 ms**	**2** nodes' ops	**45** nodes	**11** countries	**7** cloud prov.	**+1500** users



Outline

Related work
Motivation and Goals
Virtual & Invisible Private Network Design and Protocol
Evaluation
Early experiences: VIPN in the Wild
Work in Progress
References
Appendices

Work in Progress
Anonymity and Security Goals

Tamarin Formal proof approach

Security protocol **model**

Specification of the **adversary**

Specification of the protocol's **properties**

Proof that **satisfies** the properties.

Input

Output

```
rule User_S1:
  let
    ciphertext = senc(~cleartext, ~skeyUS1)
  in
  [ St_User_1( ~skeyUS1 ), Fr( ~cleartext ) ]
--[ Secret( ~cleartext, ~skeyUS1 ) ]->
  [ Out( ciphertext ) ]
```

```
rule get_skey:
  [ !SessionKey( skey, agent_1, agent_2 ) ]
--[ Reveal( skey ) ]->
  [ Out( skey ) ]
```

Base: **Dolev-Yao**

```
lemma secrecy:
  " All cleartext skeyUS1 #i #j.
      Secret( cleartext, skeyUS1 ) @ #i
    & K(cleartext) @ #j
    ==>
     (Ex #r. get_skey(skeyUS1) @ #r)
  "
```

And then, **contrapositive**...

Proof that is either Verified or Falsified

Handwritten proof approach

Algorithm 3 A sends a `'HELLO'` request to B and waits for its response.

```
1: function requestToPeer(metadata, peerConn)
2:     request ← 'HELLO'
3:     snowflake ← ⟨metadata, request⟩
4:     peerConn.sendTLS(snowflake)
5:     response ← peerConn.readSnowflake()
6:     return response
```

Algorithm 4 B receives a `'HELLO'` request from B and sends a response.

```
1: procedure receiveFromPeer(peerConn)
2:     metadata, request ← peerConn.readSnowflake()
3:     if request = 'HELLO' then
4:         peerConn.sendTLS(⟨metadata, 'ACK'⟩)
5:     else
6:         peerConn.sendTLS(⟨metadata, 'FAILURE'⟩)
```

Global Passive Adversary, *Dolev-Yao*, *Byzantine*… *polynomially bounded*, *unbounded*…

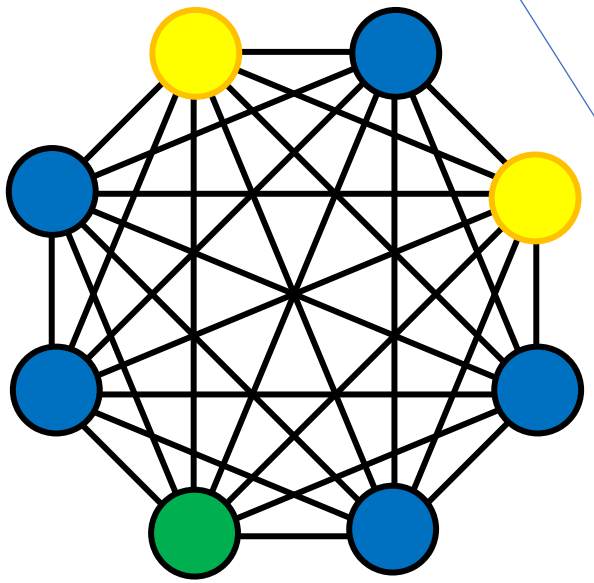
Hypotheses (e.g., with GPA):

- The network is reliable, it never fails: a packet from source A eventually arrives at destination B within a reasonable duration.
- Each peer has a public and private key pair. The private key is securely stored on the peer machine.

Lemma 1.1 (Against a GPA). *A peer A sending a `'HELLO'` request to another peer B will eventually receive an `'ACK'` response from B.*

Proof. By hypothesis, the network is reliable. Peer A sends a `'HELLO'` request that will be processed by B and will necessarily send an `'ACK'` (alg. 4, line 3) response that will eventually be received by A. □

Work in Progress
Service Goals



Snowpack node

Snowpack client

Internet service

Underlay network link

Based on [7,8]

Mininet

python™

SDN: routing supporting loops

```
graph [
directed 0
node [
  id 0
  label "node at 10.1.0.1"
  host_bandwidth_down "100 Mbit"
  host_bandwidth_up "100 Mbit"
]
edge [
  source 0
  target 0
  label "path from 10.1.0.1 to 10.2.0.2"
  latency "10 ms"
  jitter “1 ms"
  packet_loss 0.2
]
]
```

Shadow-GML graph format: underlay network

- **Link** and **nodes** characteristics variations
- **Number of users/nodes** ; ratio.
- User **route choices** (random/arbitrarily chosen)
- **Scenarios to be tested**: Internet surfing, video streaming, real-time communication, synthetic scenarios, HTTP1/2, QUIC…
- **Comparison** with other competitors

Remaining trusted third-parties

- Nodes' directory
- Users' directory
- Operators

Resistance to « degradation » attacks

- Traffic analysis
- Denial-of-service

Anonymous Access control

Monitoring & indicators sharing without degrading privacy

Legal Compliance & Open source

End-to-end Mode evaluation

snowpack LIP6 cnrs SORBONNE UNIVERSITÉ université PARIS-SACLAY cea

Thank you for your attention

Any questions?

References

[1] D. L. Chaum, *Untraceable electronic mail, return addresses, and digital pseudonyms*, Commun. ACM, vol. 24, no. 2, pp. 84–90, Feb. 1981, doi: 10.1145/358549.358563.

[2] F. Shirazi, M. Simeonovski, M. R. Asghar, M. Backes, and C. Diaz, *A Survey on Routing in Anonymous Communication Protocols*, ACM Comput. Surv., vol. 51, no. 3, p. 51:1-51:39, Jun. 2018, doi: 10.1145/3182658. Available at: https://lirias.kuleuven.be/bitstream/123456789/626536/2/a51-shirazi.pdf.

[3] J. van den Hooff, D. Lazar, M. Zaharia, and N. Zeldovich, *Vuvuzela: scalable private messaging resistant to traffic analysis*, in Proceedings of the 25th Symposium on Operating Systems Principles, in SOSP '15. New York, NY, USA: Association for Computing Machinery, Oct. 2015, pp. 137–152. doi: 10.1145/2815400.2815417.

[4] N. Tyagi, Y. Gilad, D. Leung, M. Zaharia, and N. Zeldovich, *Stadium: A Distributed Metadata-Private Messaging System*, in Proceedings of the 26th Symposium on Operating Systems Principles, in SOSP '17. New York, NY, USA: Association for Computing Machinery, Oct. 2017, pp. 423–440. doi: 10.1145/3132747.3132783.

[5] N. Gelernter, A. Herzberg, and H. Leibowitz, *Two Cents for Strong Anonymity: The Anonymous Post-office Protocol*, in Cryptology and Network Security, S. Capkun and S. S. M. Chow, Eds., in Lecture Notes in Computer Science. Cham: Springer International Publishing, 2018, pp. 390–412. doi: 10.1007/978-3-030-02641-7_18.

[6] A. Kwon, H. Corrigan-Gibbs, S. Devadas, and B. Ford, *Atom: Horizontally Scaling Strong Anonymity*, in Proceedings of the 26th Symposium on Operating Systems Principles, in SOSP '17. New York, NY, USA: Association for Computing Machinery, Oct. 2017, pp. 406–422. doi: 10.1145/3132747.3132755.

[7] A. H. Kwon, D. Lazar, S. Devadas, and B. Ford, *Riffle: An efficient communication system with strong anonymity*, Proc. Priv. Enhancing Technol., 2015, doi: 1721.1/128773.

[8] D. Chaum et al., *cMix: Anonymization by high-performance scalable mixing*, in USENIX Security, 2016. Available at: https://chaum.com/wp-content/uploads/2022/02/Chaum-cMix-Anonymization-by-High-Performance-Scalable-Mixing.pdf

References

[9] A. M. Piotrowska, J. Hayes, T. Elahi, S. Meiser, and G. Danezis, *The Loopix Anonymity System*, presented at the 26th USENIX Security Symposium (USENIX Security 17), 2017, pp. 1199–1216. Available at: https://www.usenix.org/conference/usenixsecurity17/technical-sessions/presentation/piotrowska.

[10] D. M. Goldschlag, M. G. Reed, and P. F. Syverson, *Hiding Routing information*, in Information Hiding, R. Anderson, Ed., in Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 1996, pp. 137–150. doi: 10.1007/3-540-61996-8_37.

[11] R. Dingledine, N. Mathewson, and P. Syverson, *Tor: The Second-Generation Onion Router*, Defense Technical Information Center, Fort Belvoir, VA, Jan. 2004. doi: 10.21236/ADA465464. Available at: https://spec.torproject.org/tor-design.

[12] I. Clarke, O. Sandberg, B. Wiley, and T. W. Hong, *Freenet: a distributed anonymous information storage and retrieval system*, in International workshop on Designing privacy enhancing technologies: design issues in anonymity and unobservability, Berlin, Heidelberg: Springer-Verlag, Jan. 2001, pp. 46–66. doi: 10.1007/3-540-44702-4_4.

[13] K. Bennett and C. Grothoff, *GAP–practical anonymous networking*, in International Workshop on Privacy Enhancing Technologies, Springer, 2003, pp. 141–160. doi: 10.1007/978-3-540-40956-4_10.

[14] Q. Wang and N. Borisov, *Octopus: A Secure and Anonymous DHT Lookup*, in 2012 IEEE 32nd International Conference on Distributed Computing Systems, Macau, China: IEEE, June 2012, pp. 325–334. doi: 10.1109/ICDCS.2012.78.

[15] H. Corrigan-Gibbs and B. Ford, *Dissent: accountable anonymous group messaging*, in Proceedings of the 17th ACM conference on Computer and communications security - CCS '10, Chicago, Illinois, USA: ACM Press, 2010, p. 340. doi: 10.1145/1866307.1866346.

[16] B. Zantout and R. Haraty, *I2P data communication system*, in Proceedings of ICN, Citeseer, 2011, pp. 401–409.

[17] R. Sherwood, B. Bhattacharjee, and A. Srinivasan, *P5: a protocol for scalable anonymous communication*, J. Comput. Secur., vol. 13, no. 6, pp. 839–876, Dec. 2005, doi: 10.3233/JCS-2005-13602.

[18] R. Shokri, N. Yazdani, and A. Khonsari, *Chain-Based Anonymous Routing for Wireless Ad Hoc Networks*, in 2007 4th IEEE Consumer Communications and Networking Conference, Las Vegas, NV, USA: IEEE, Jan. 2007, pp. 297–302. doi: 10.1109/CCNC.2007.65.

References

[19] A. Pfitzmann and M. Hansen, *A terminology for talking about privacy by data minimization: Anonymity, Unlinkability, Undetectability, Unobservability, Pseudonymity, and Identity Management*. Aug. 2010. Available at: http://dud.inf.tu-dresden.de/literatur/Anon_Terminology_v0.34.pdf.

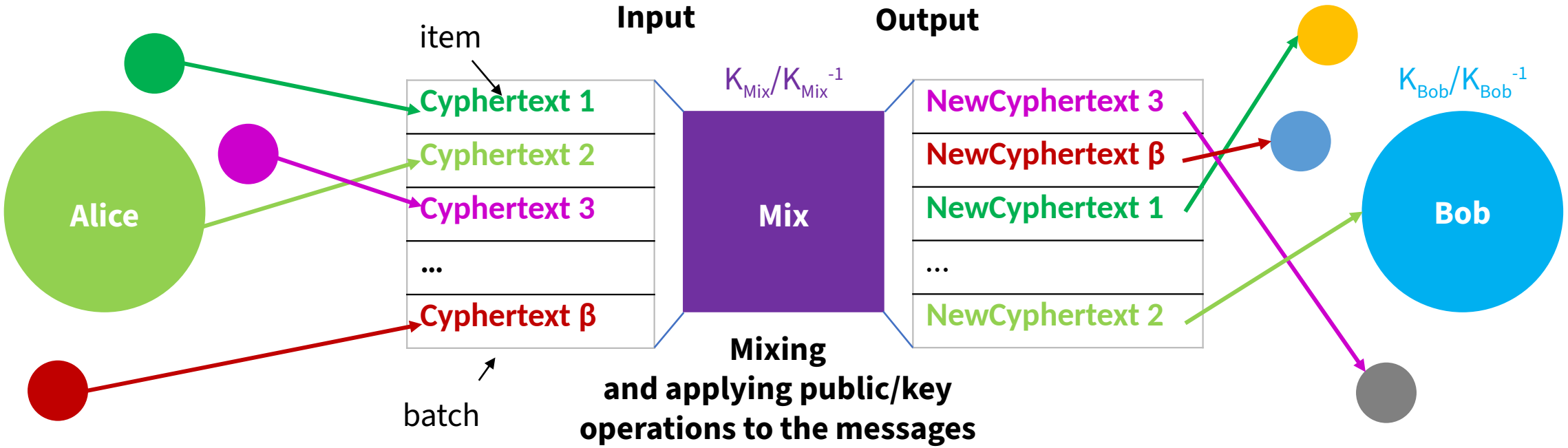
snowpack LIP6 cnrs SORBONNE UNIVERSITÉ université PARIS-SACLAY cea

Appendices

David Chaum's seminal paper

Asymmetric cryptographic operations **only**.

Usage : End-to-End



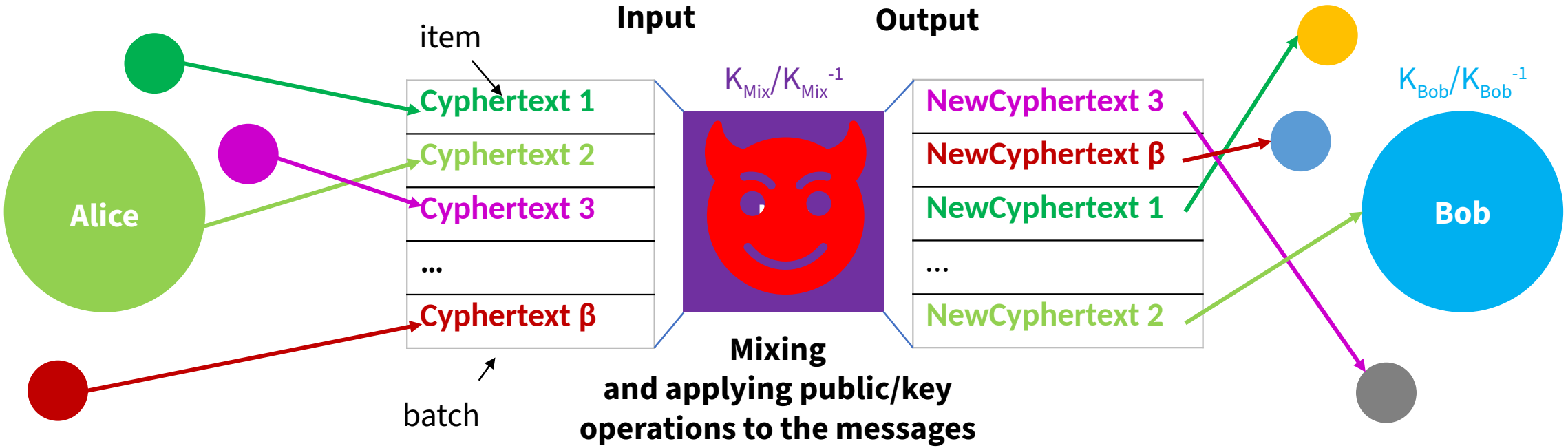
M: a message.
K_A : public key of A.
K_A^{-1} : private key of A.
$K_A^{-1}(K_A(M)) = (K_A(K_A^{-1}(M)) = M$

To avoid guessing $Y = M$ by testing $KA(Y) = KA(M)$, a large string of random bits R is concatenated with M before encryption: M is encrypted using $KA(R, M)$.

David Chaum's seminal paper

Asymmetric cryptographic operations **only**.

Usage : End-to-End



$$K_{Mix}\,(R_{Mix}, K_{Bob}(R_{Bob}, M), Bob)$$

Cyphertext 2

Anonymity lost

$$K_{Bob}(R_{Bob}, M), Bob$$

NewCyphertext 2