Drac: An architecture for Anonymous Low-Volume Communications

G. Danezis (Microsoft Research Cambridge),
C. Diaz, **C. Troncoso (KU Leuven/COSIC)**,
B. Laurie (Google Inc)

Introduction

- Traffic data of real time communications leaks information
  - Timing (military actions), volume (strength of relationships), participants (medical status),....
- Few systems provide anonymity against global passive adversary for real time communications
  - Conceal patterns entails high cost (e.g., bandwidth peaks in web traffic)
- What if the application requires limited bandwidth or regular traffic (VoIP, IM)?
  - Padding to destroy traffic patterns becomes viable

Drac: architecture and goals

- Friend-of-a-friend architecture
  - Better scalability
  - Sybil prevention
  - Build incentives
  - Stable anonymity sets

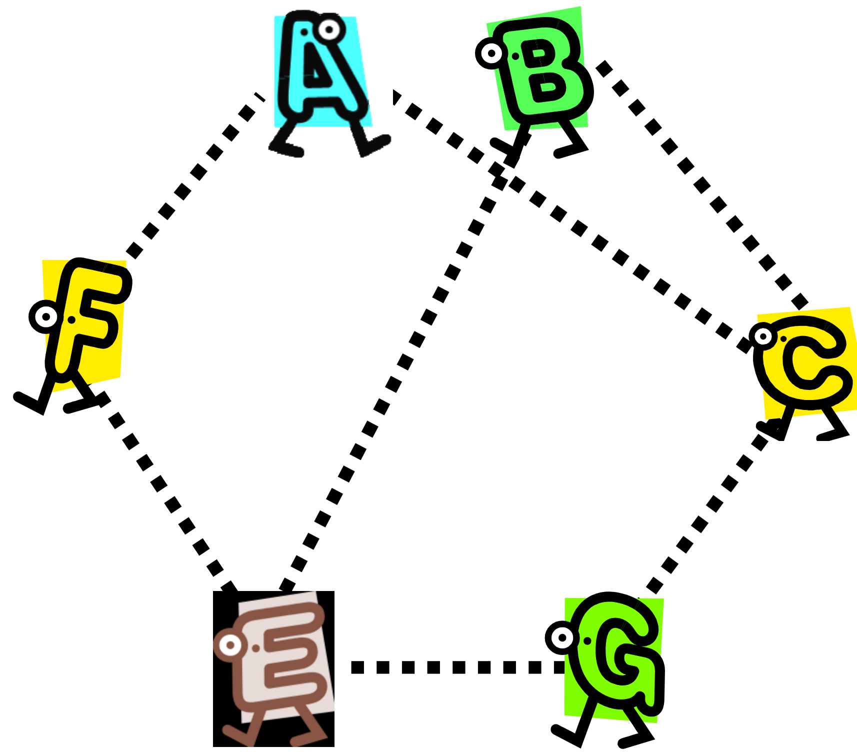
- UNOBSERVABILITY of communication between friends
  - The adversary cannot tell whether they speak at all
- ANONYMITY of other relationships
  - The adversary cannot find further contacts

Relationships in Drac

- **Friends**
  - Trusted
  - Visible to the attacker
  - Unobservable communications
- Contacts
  - Not trusted
  - Not known to the adversary
  - Relationship confidentiality
- Private Presence Server
  - “Rendez-vous” to find contacts

Heartbeat connections

- Between each pair of friends
- Signaling
  - presence to friends
  - establish communications
  - communicate with Presence Server
- Continuous traffic
  - very low bandwidth
  - bidirectional
- **No** additional info to the adversary, "public" information



Small remarks

REMARK 1

In the rest of the talk I will ignore cryptographic aspects of the protocols as well as key management.
Details in the paper

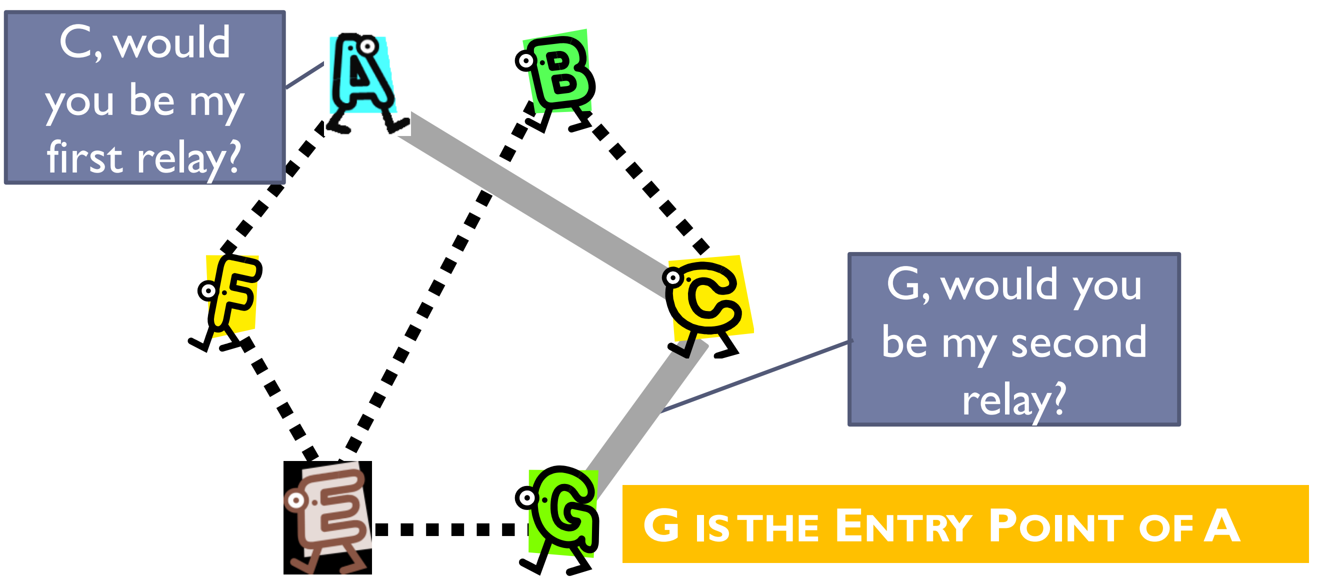
REMARK 2

In the rest of the talk I assume that all connections are padded, i.e., they carry constant traffic to counter traffic analysis

Entry points

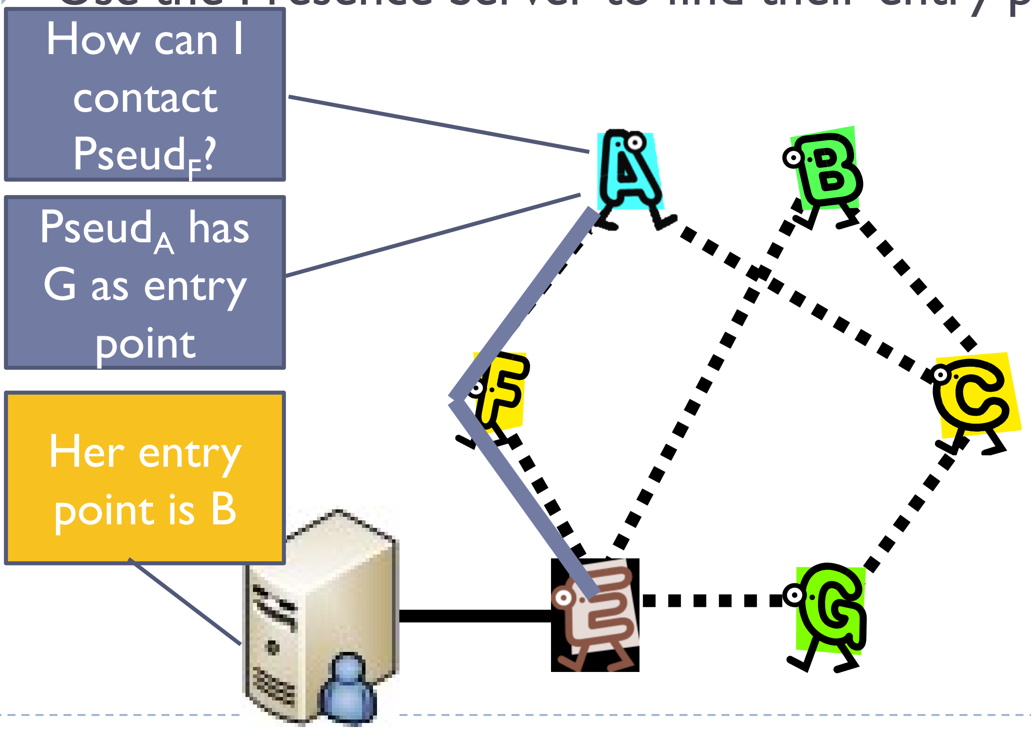
- Direct communications reveals the identity of participants
- ENTRY POINT: proxy D hops away from user
  - **Every** user has an entry point
  - ...even if they don't want to start a conversation! (for other users to find them and to provide unobservability)

D=2



Finding contacts

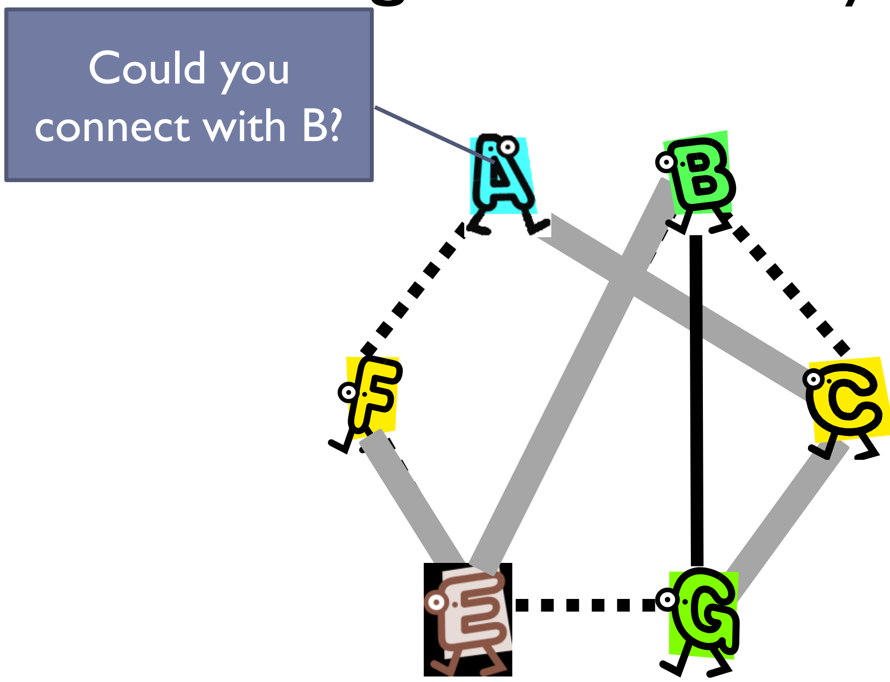
- If Alice wants to speak with her friends she knows where they are
  - Choose them as first hop in the circuit to entry point
- What about contacts?
  - Use the Presence Server to find their entry points



1. Construct circuit to PS over heartbeat channels
2. Send entry point to PS under a pseudonym
   - PS does not learn who and where is A
3. Ask for entry point of conversation partner
   - Presence server **cannot** learn who issued the request!
   - nor who is the conversation partner

Establishing communications with contacts

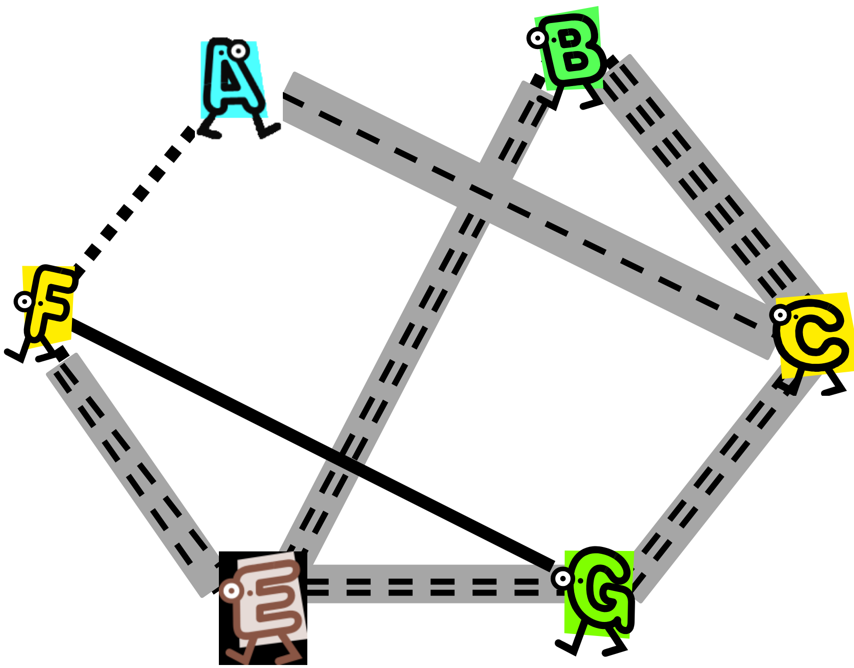
- From the example before…
  - A's entry point is G, and F's entry point is B
- Establish a **bridge** between entry points



Communication channel between A and F: A-C-G-B-F

Epochs in Drac

- Creating and tearing down circuits reveals information
  - Synchronous start and end of communications: EPOCHS
  - Epoch prepared in previous epoch



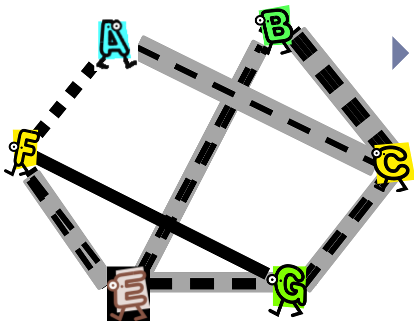
- Circuits:
  - A-C-G
  - B-C-B
  - C-G-E
  - G-E-F
  - E-B-C
  - F-E-B
- Conversations

  A speaks to G (connect G and F)

  F speaks to B (no bridge!)

Contact communication anonymity

- Assume all bridges and circuits per link are observable… what can the adversary do?



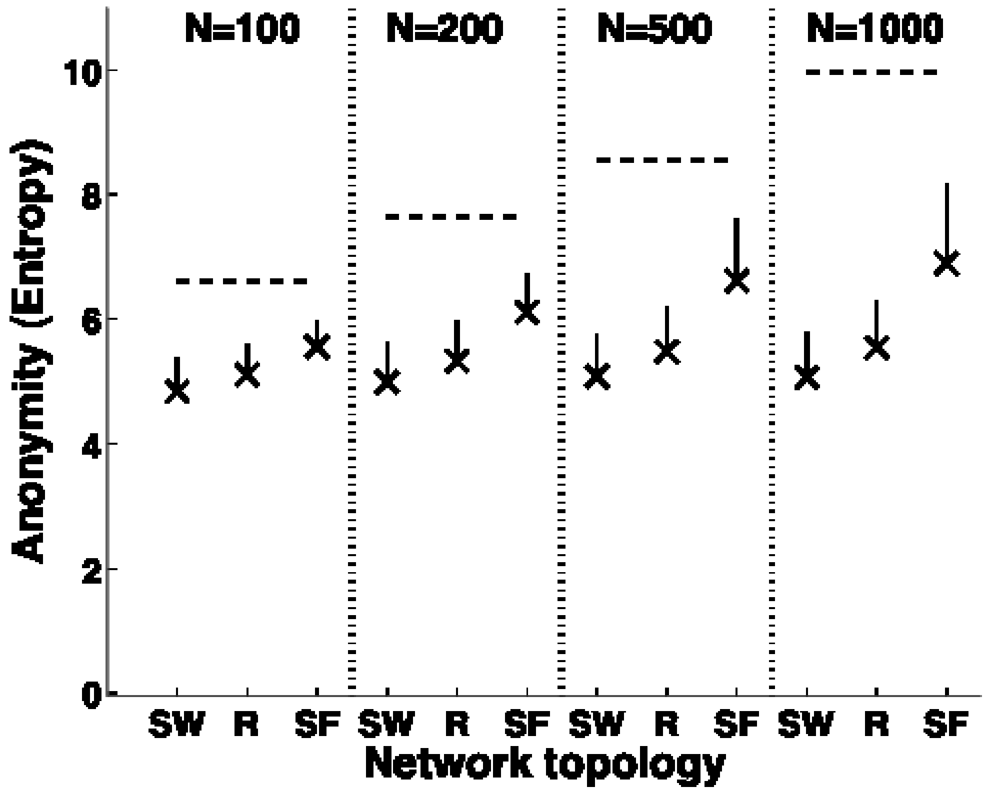
- Could have been...
  - A-C-G, B-C-B, C-G-E, G-E-F, E-B-C, F-E-B
  - A-C-G, B-C-G, C-B-C, G-E-F, E-B-E, F-E-G

- No certainty that A is communicating...
  - Usual anonymity metrics are not straight forward to compute
  - We evaluate anonymity of each half of circuit separately, starting from bridge (**no** end-to-end anonymity)
    - by checking all paths that lead to each of the initiators

- In the paper we also analyse anonymity towards the presence server

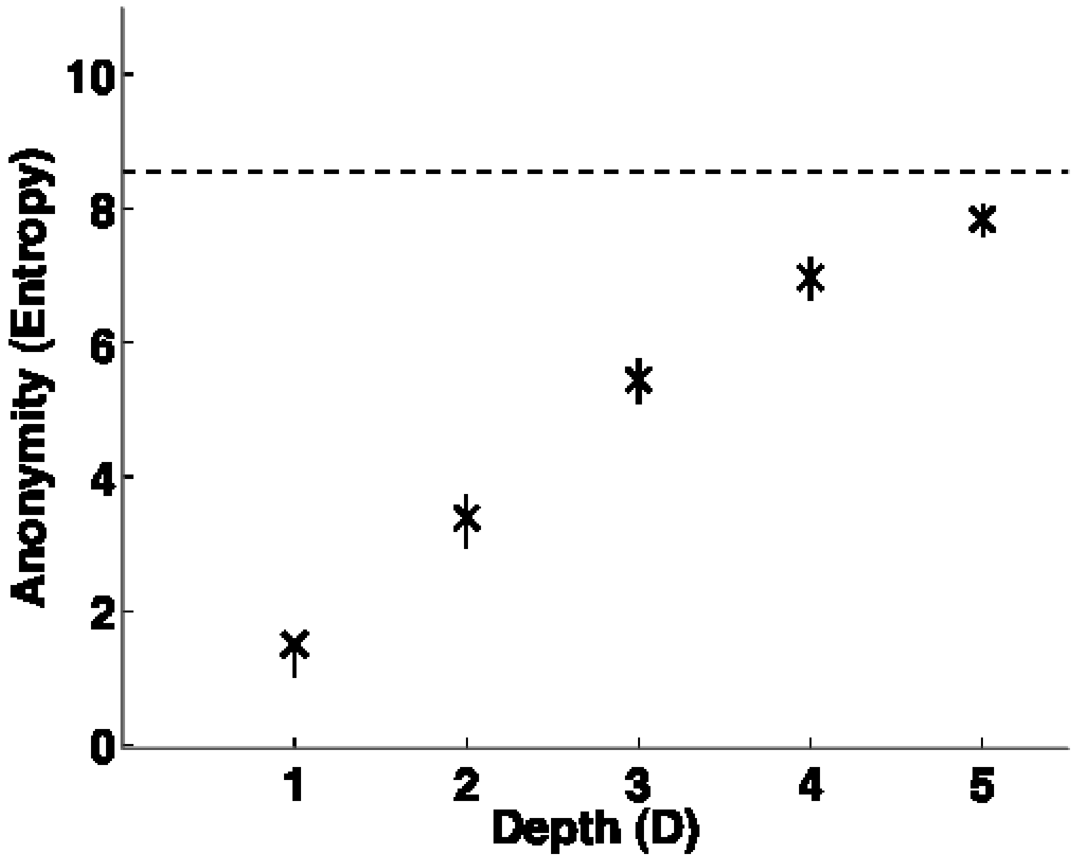
Results: topology

- Three topologies: small-world, scale-free, random



Parameters: 10 friends, D = 3

Results: circuit depth

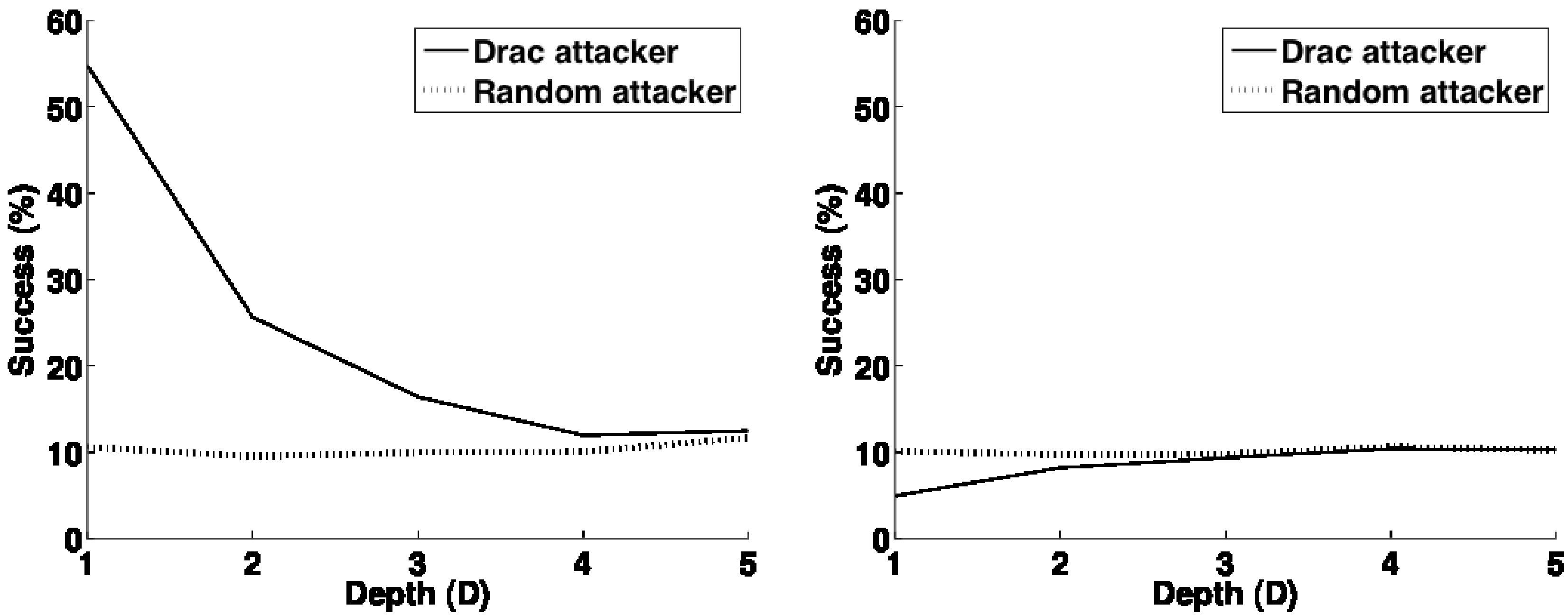


Parameters: SW net, N = 500, 10 friends

Unobservability

- Communications with friends: fully unobservable
- Communications with contacts: bridges observable
  - X : total nr of contact communications (assume known by adversary)
- Evaluation:
  1. Adversary constructs set S with top 2X users (highest probability of having created a bridge)
  2. Random adversary: constructs set R with 2X random users
  3. Select user u_A who *is* communicating with a contact
     - Test adversaries success (u_A in S? and u_A in R?)
  4. Select user u_Z who *is not* communicating with a contact
     - Test adversaries success (u_Z in S? and u_Z in R?)

Results



Parameters: SW net, N = 500, 10 friends, C = 25

Conclusions

- Low bandwidth applications allow for connections padding to prevent traffic analysis
- Hiding friends is hopeless, leverage to achieve anonymity of further relationships
  - And provide unobservability of communications with friends
- Friend of friend architecture
  - Scalability, incentives, avoid sybil attacks, stable anonymitysets
- Depth of circuit is a security parameter
  - but anonymity also depends on the mixing properties of the social graph

Open questions

- The design seems promising…
  - We only analyzed one epoch
    - Intersection attacks
    - Optimal duration security vs usability
  - We did not compute end to end anonymity
    - MCMC for proper computation of probability distributions
  - Unobservability metrics,
  - Deniability?
  - Resistance to corrupted nodes
  - Social network dynamics
  - ….

Questions?

1. What the *%&#" is Drac?

Onion encryption

$$u_X \rightarrow u_Y \rightarrow u_Z \Rightarrow u_U \rightarrow u_V \rightarrow u_W$$

$$u_X \rightarrow u_Y : E_{k_{XY}}(E_{k_{XZ}}(E_{k_{XW}}(M)))$$

$$u_Z \Rightarrow u_U : E_{k_{XW}}(M)$$

$$u_V \rightarrow u_W : E_{k_{VW}}(E_{k_{UW}}(E_{k_{XW}}(M)))$$

Private presence server

- Private Presence server: Honest but curious
- There could be several of them
- User u_A has long-term identifier ID_A (user may have several, one per circle of contacts, so they cannot find out they know the same user)
- Contacts A and B share a key K_{AB}

Presence

- unlinkability between time periods (epochs), avoid long-term pseudonymous profiling: "*id du jour*" IDJ
- T published by Presence server

$$IDJ_A = H(T, ID_A)$$

- B sends this message to the PS:

$$E_{PK_{PS}}(IDJ_A, E_{K_{AB}}(E_B, g^{r_B}))$$

- If A wants to talk to B, she sends g^{r_A} to E_B (next epoch)
- session key: $k_{AB} = g^{r_A r_B}$
- update long term key: $K'_{AB}=H(k_{AB},K_{AB})$

Experimental setup

- Simulator implemented in python
- Topologies: small world, scale free, random
  - f friends on average (selected according to topology)
  - f randomly selected contacts
- Single epoch per experiment (no multiple epoch analysis)
  - heartbeat connections: between friends, and between end of presence circuit and presence server
  - communication circuits and bridges; adversary can see nr of circuits per link and distinguish bridges
  - 10% of users communicating with contacts (randomly selected)
- One sample per experiment:
  - contact communication anonymity
  - presence anonymity
  - contact communication unobservability

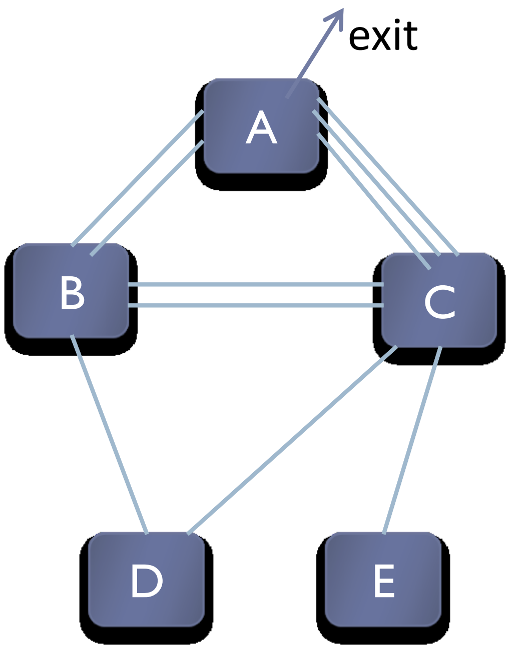
Anonymity towards the presence server

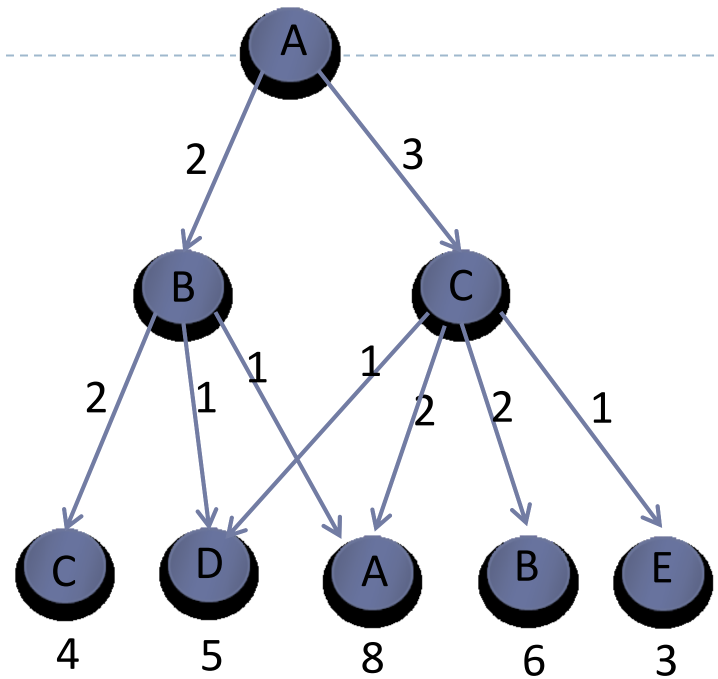
- start from connection to Presence Server (end of circuit)
- check all paths that lead to each of the initiators

$$\Pr_i[E_{PA}] = \frac{P_i}{\sum P_j}, 1 \le i \le N$$

$$H_A = -\sum_{i=1}^{N} [E_{PA}]\log_2 \Pr_i[E_{PA}]$$

Example



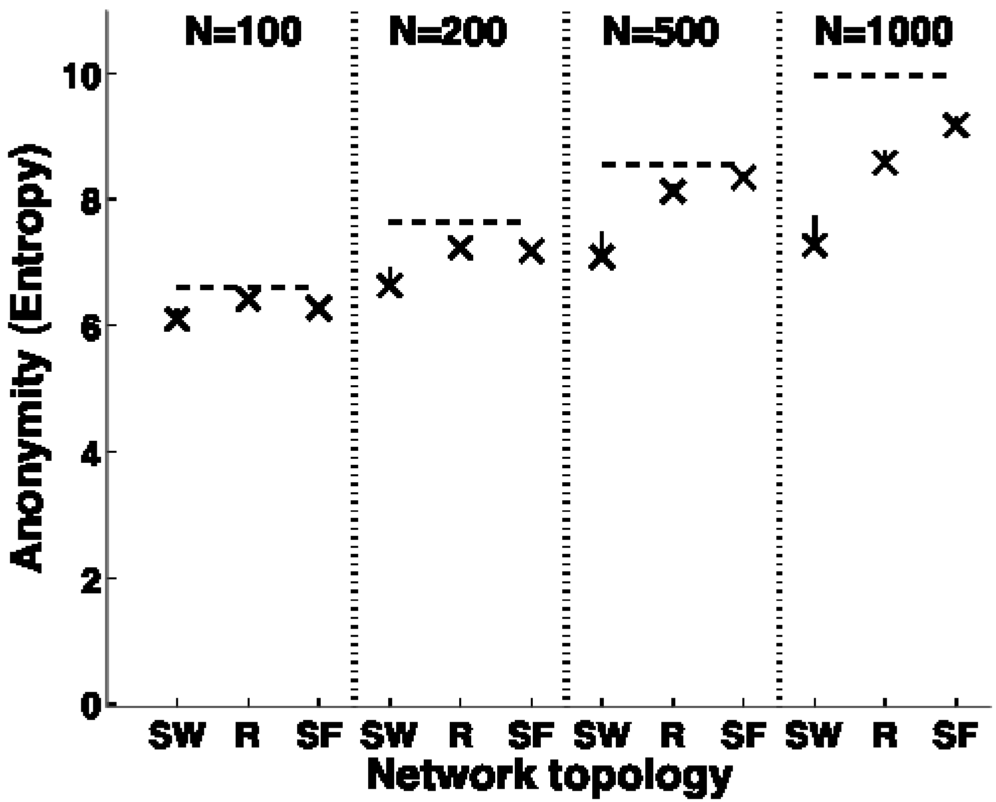


true paths:
- A-C-B
- B-C-A
- C-A-B
- D-B-A
- E-C-D

possible paths:
- C-B-A (x4)
- D-B-A (x2)
- A-B-A (x2)
- D-C-A (x3)
- A-C-A (x6)
- B-C-A (x6)
- E-C-A (x3)

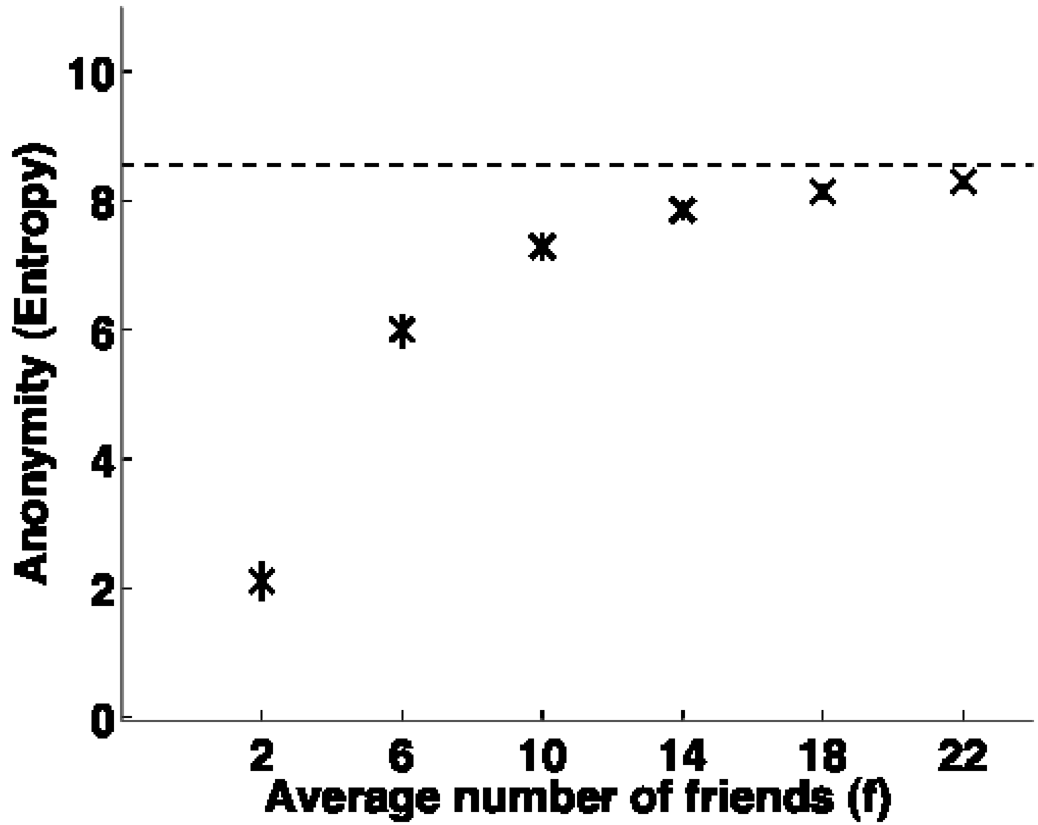
Prob (caller, exit A):
- Pr(A) = 8/26 = 0,3
- Pr(B) = 6/26 = 0,23
- Pr(C) = 4/26 = 0,15
- Pr(D) = 5/26 = 0,19
- Pr(E) = 3/26 = 0,12

Results: Topology



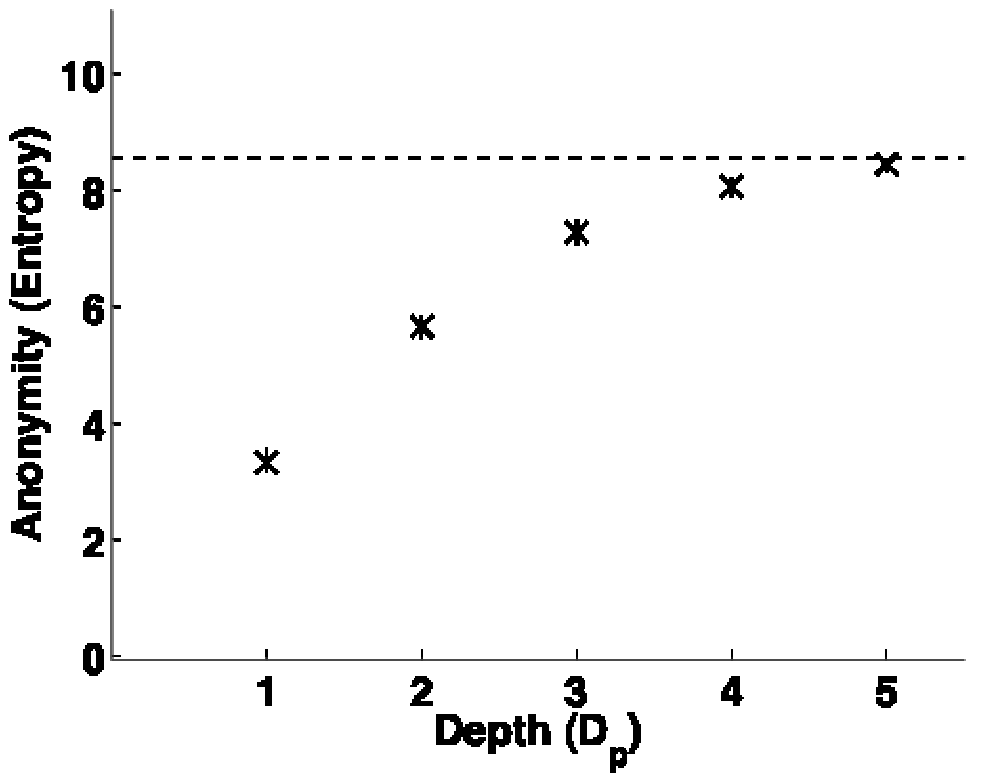
Parameters: 10 friends, D_p = 3

Results: number of friends



Parameters: SW net, N = 500, D_p = 3

Results: circuit depth



Parameters: SW net, N = 500, 10 friends