Ghost Train for Anonymous Communication

Przemysław Błaśkiewicz, Mirosław Kutyłowski,
Jakub Lemiesz, Małgorzata Sulkowska

Wrocław University of Science and Technology, Poland

SpaCCS 2016, Zhangjiajie

Protection of traffic data

Confidentiality

- **encryption:** easy to hide the contents of messages exchanged over public networks
- **traffic volume:** hard to hide
  – a dummy traffic is only a partial solution

Communication management

- **communication protocols:**
  - the destination address almost always given explicitly
  - the source address frequently given but not authenticated
- **routing protocols:** oriented on efficiency and not data protection

Who is talking with whom

information on who is talking with whom is crucial

Law enforcement:

- *May 11, 2014: Speaking at a debate in April, former intelligence boss and retired Gen. Michael Hayden admitted the NSA uses metadata to "kill people."*

  metadata are (sender,recipient) data, and not the communication contents

- forensics: connection data used to **deanonymize**

Dark side of traffic analysis

- **business intelligence**
  unfair competition, business attacks
  creation of **monopolies** based on access to data and not on production/services quality
- **organized crime**
  traffic data may **ease committing crime** and reduce the risks
- **national security**
  it is not only security agencies that may use traffic data
  the terrorists might be more advanced in this field
  - higher budget
  - highly paid specialists
  - no legal limitations (e.g. personal data protection rules)

Broadcast

Broadcasting based protection

- broadcast data **to all**
  e.g. radio, satellite transmission
- **anybody** in the transmission range can be the recipient

Problems

- careful choice of the encryption scheme
  e.g. RSA hybrid encryption is not applicable
- the sender is not protected
- if the communication is bidirectional, one may derive (sender,receiver) candidate pairs
- costly method, limited bandwidth

Token ring

Token ring

- the servers organized in a **single ring**
- each encrypted message travels **around the whole ring**
- the recipient of the message can see the ciphertext while forwarding
  perfect destination anonymity

Problems

- **lack of scalability**
- communication latency
- no sender anonymity

Onion routing

Onion routing technique

- a message sent **over a random path to its destination**
- the encryption method guarantees that **an intermediate node learns only the next node** on the path
  (the previous node learnt too)
- the encryption method guarantees that cryptanalytic **linking of incoming and outcoming messages infeasible** for an observer

Problems

- based on **node mixing**: **many messages must be processed by a node at the same time**
  otherwise easy to recover the path based on time sequence
- does not hide the senders and the receivers
- **security proofs** concern only an adversary that can see the message on the communication links **but not their transmitting time**

TOR

TOR system

- a system implementing onion routing,
- connection based protocol:
  - first **a connection is built using Onion Routing like technique**
  - when the connection is established, then a transmission starts
  - on intermediate nodes: ciphertext change (symmetric method)
  - intermediate nodes learn only the neighbors on the path

Problems

- when a transmission terminates, then traffic decreases on the whole path
- **if only one link broken at a time, then connection visible for a passive adversary**
- moreover: **it suffices to monitor the source and the destination servers** only

Privacy challenge

How to secure the traffic against traffic analysis?

Practice

Existing tools give only a **basic protection**, ineffective against a powerful adversary.

The users (e.g. of TOR) may falsely assume that they are anonymous.

Theory

there is no good theoretical solution so far.
(the situation much different from, say, the state-of-the-art in encryption technologies)

Beimel-Dolev Busses

Amos Beimel and Shlomi Dolev. Buses for anonymous message delivery. J. Cryptology, 16(1):25-39, 2003.

Bus

- many ***seats***
- each seat can hold a **single ciphertext**
- a bus **travels through the network**
- when a bus reaches the destination of a ciphertext, the destination node can decrypt it and understand

Buses

Versions

- for a **token ring** of N nodes: $N(N-1)/2$ **seats**, each seat for a pair of nodes
- for **random walks**: **no assigned seats**, the number of seats much smaller, if no free seat then **overwriting a random seat**
- a combination of many intersecting rings: transfer between busses

Problems

- some care necessary when choosing encryption method (ciphertext properties may betray the destination)
- either a huge bus or overwriting possible
- well defined network needed or random walks
- random walk works well for certain graphs (with expander properties)
- **hiding senders requires inserting a fake ciphertext** (taking a seat)

Drunk Motorcyclist

Adam Young and Moti Yung. The drunk motorcyclist protocol for anonymous communication. IEEE Conference on Communications and Network Security, 2014.

Strategy

- drunk motorcyclist performs **a random walk** over the network
- motorcycle carries **a single ciphertext**, **never changed** after sending
- **a counter (time to die)** - each motorcyclist makes a fixed number of steps
- the ciphertext is **sent many times** – necessary to make sure that some ciphertext arrives at its destination

Problems

- in order to hide sending activity one has **to sent drunk motorcyclist at each moment**.
- the counter **betrays the senders** even if no full view of the network

Ghost Train
architecture

A ghost train

1. performs a **random walk** through the network
2. holds a **long bit array**

A network node

upon ghost train arrival:

1. **derives messages** (if any) from the bit array
2. **changes** some number of **bits** in the bit array
   - in order to **encode** a message for someone
   - or to **hide** sender's inactivity

Ghost train
information fading

The classical concept

- in a data stream **each information has an assigned place**
- **overwriting destroys completely the old information** and puts the new one on this place

The ghost data concept

- the places assigned to different informations **overlap**
- through overwriting the **old data fade** and **eventually disappear**
- one overwriting operation affects many old data but each of them only slightly
  ... with high probability

Comparison

classical approach: message **intact or completely lost**

ghost approach: messages **decay over the time**

Ghost train
encoding details

Communicating nodes

- nodes x, y **share a key** K_{xy}

Data array $P.B$ carried by a ghost train P

- $P.B$ has length n, it **always stores** $n/2$ **ones and** $n/2$ **zeroes**
- x encodes a bit b in $P.B$ by setting the contents of $P.B$ to b at the **pseudorandom positions** derived with the secret K_{xy}
- for the sake of **balance** of zeroes and ones, some changes on other random positions
- $P.B$ is a kind of a **Bloom filter**

Ghost train
encoding details

Train P metadata

- identifier $P.id$
- history $P.H$ – the list of recent nodes visited by the train P

time divided into **epochs** – within an epoch the same bit is sent

Ghost train
encoding

Arguments: epoch index i, slot index t, list of destinations L, received train P

1 inspect the history $P.H$ and determine node $y \in L$ for which train P has not been used in epoch i and bit to be sent to y

2 **if** $y \neq null$ **then**

3 $\quad S \leftarrow \mathbf{Hash}(K_{xy}, i, t, P.id)$

4 **else**

5 $\quad$ choose S and bit **at random**;

6 $ones \leftarrow$ **the number of ones on positions** S in $P.B$;

7 **set all positions from** S in $P.B$ to bit ;

8 **if** $bit = 1$ **then**

9 $\quad ones \leftarrow k - ones$**;**

10 **while** $ones > 0$ **do**

11 $\quad r \leftarrow rand[1, n]$;

12 $\quad$ **if** $r \notin S \wedge P.B[r] = bit$ **then**

13 $\quad\quad P.B[r] \leftarrow 1 - bit$;

14 $\quad\quad ones \leftarrow ones - 1$;

15 **push**$((x, t), P.H)$; /* push (x, t) to 1st position in $P.H$ */

Ghost train
encoding

properties observed

Bit hiding

- some number of 1's changed to 0,
- the same number of 0's changed to 1
- ... so it is not observable which bit has been encoded

Location hiding

- about half of the positions of S is not observable – they already contain the right bit
- lack of this information make **cryptanalysis (e.g. brute force) much harder**

Ghost train
decoding procedure

Arguments: i, t, L, P

1 **foreach** *node* $y \in L$ **do**

2 **foreach** *position* j *of an entry* (y, τ) *in* $P.H$ **do**

3 **if** $j \geq t$ **then** $r \leftarrow 1$;

4 **else** $r \leftarrow 2$;

5 ;

6 $S_r \leftarrow h(K_{yx}, i - 2 + r, \tau, P.id)$;

7 $y.X_r \leftarrow y.X_r +$ the number of ones in $P.B$ on positions from S_r

  ;

8 $y.b_r \leftarrow y.b_r + k$;

idea: count the number of ones in the areas assigned to transmitted bit

final result: statistics over the whole epoch, majority voting

Passive observer

Indistinguishability

- a **passive adversary** that observes the ghost train entering a node and leaving the node
- **Indistinguishability Game:**
  two options
  1. the protocol executed
  2. the derivation of S replaced by random choice
- **argument:**
  - if the adversary cannot distinguish between both options, then the same attack advantage for the 2nd option
  - 2nd option: no advantage for the adversary
  - distinction between the options $\leftrightarrow$ PRNG is cryptographically weak

Malicious nodes

Possibilities

- **impossible to change many bits** (without being detected)
- infeasible to “attack” positions corresponding to one message – **the positions are unknown**

Message lifetime

Challenge

assume that the ghost train has done T hops after leaving x until it reaches y

what is the probability that the large majority of k positions communicated to y has the value set by x?

Goal

for s packets received, $\mathbb{X}$ standing for the number of positions, where the original value survives:

$$\Pr[\mathbb{X} \geq f \cdot s \cdot k] \geq 1 - \delta \quad \text{and} \quad \mathbb{P}[\mathbb{X} \leq (1 - f) \cdot s \cdot k] \leq \delta$$

Message lifetime

Road map

- for a given position derive the expected value of the bit stored there after T hops
- estimate the variance of this random variable
- derive the probability for correct decoding

quite tight analytic results in the paper

Example parameter settings

example

- N – the network size
- epoch length $T = \left\lceil \sqrt{N \log N} \right\rceil$
- Bloom filter length $n \sim \left\lceil \sqrt{N \log N} \log N \right\rceil$

then

- **probability of successful message delivery**

$$\sim 1 - \frac{1}{N}$$

discussion

- the epoch is relatively long in order to guarantee message delivery and high security margin
- filter size is not a big issue (even for millions of nodes)

Conclusions

Achieved

- fully oblivious routing
- encryption method for multiple ciphertexts that does not assign separate locations for each ciphertext
- full control over the communication volume, no dummy messages
- **hard to perform denial of service and kill selectively ciphertexts from a given node**

Challenges

- efficiency of communication (random walk)
- channel bandwidth

Lessons learnt

Application for malicious purposes

- a growing threat of communication over the public network for evil purposes
- e.g. between malware
- random walks do not betray communicating parties
- feasibility of limited bandwidth communication
- ciphertexts need not to have fixed location and can be hidden in noise

Thanks for your attention!

Contact data

1. Miroslaw.Kutylowski@pwr.edu.pl
2. http://kutylowski.im.pwr.edu.pl
3. http://cs.pwr.edu.pl