# Website Fingerprinting

Attacking Popular Privacy Enhancing Technologies with the Multinomial Naïve-Bayes Classifier

**Dominik Herrmann**, Hannes Federrath
University of Regensburg, Germany

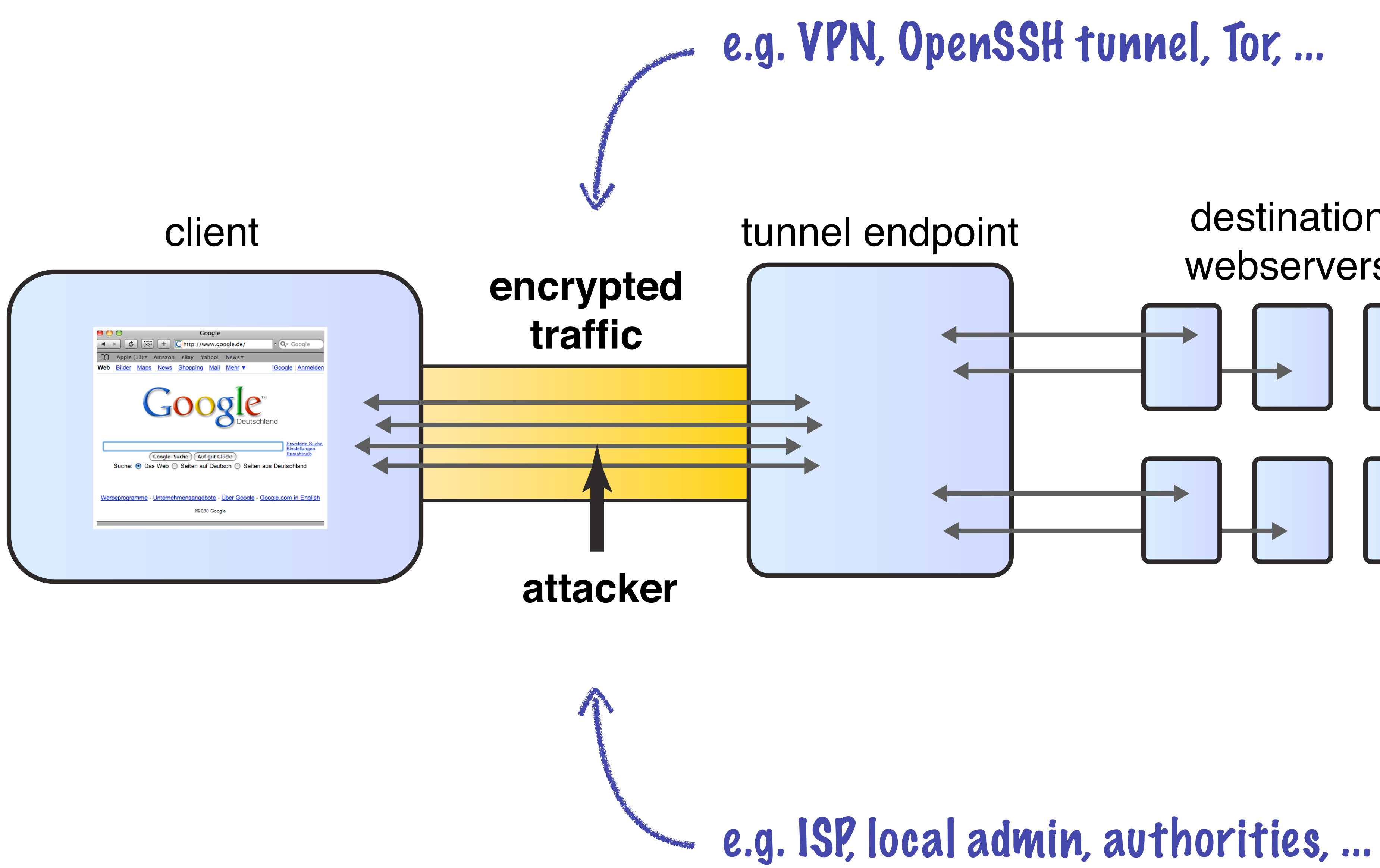
Rolf Wendolsky
JonDos GmbH

# Motivation – To Whom It May Concern

- Various **Privacy Enhancing Technologies (PET)** offer protection against eavesdropping
  - SSH/SSL tunnels and VPNs
  - multi-hop anonymisation services

- Users want protection against malicious ISPs and other users
- Criminals want to hide their activities from the authorities

# Attack Scenario



# Overview of Our Fingerprinting Attack

- Attacker wants to learn URLs of websites that are requested over an encrypted tunnel by the victim.
- **Website Fingerprints:** Attack exploits characteristic structure of websites.
- **Attacker:** passive, local, external observer

## PROCEDURE

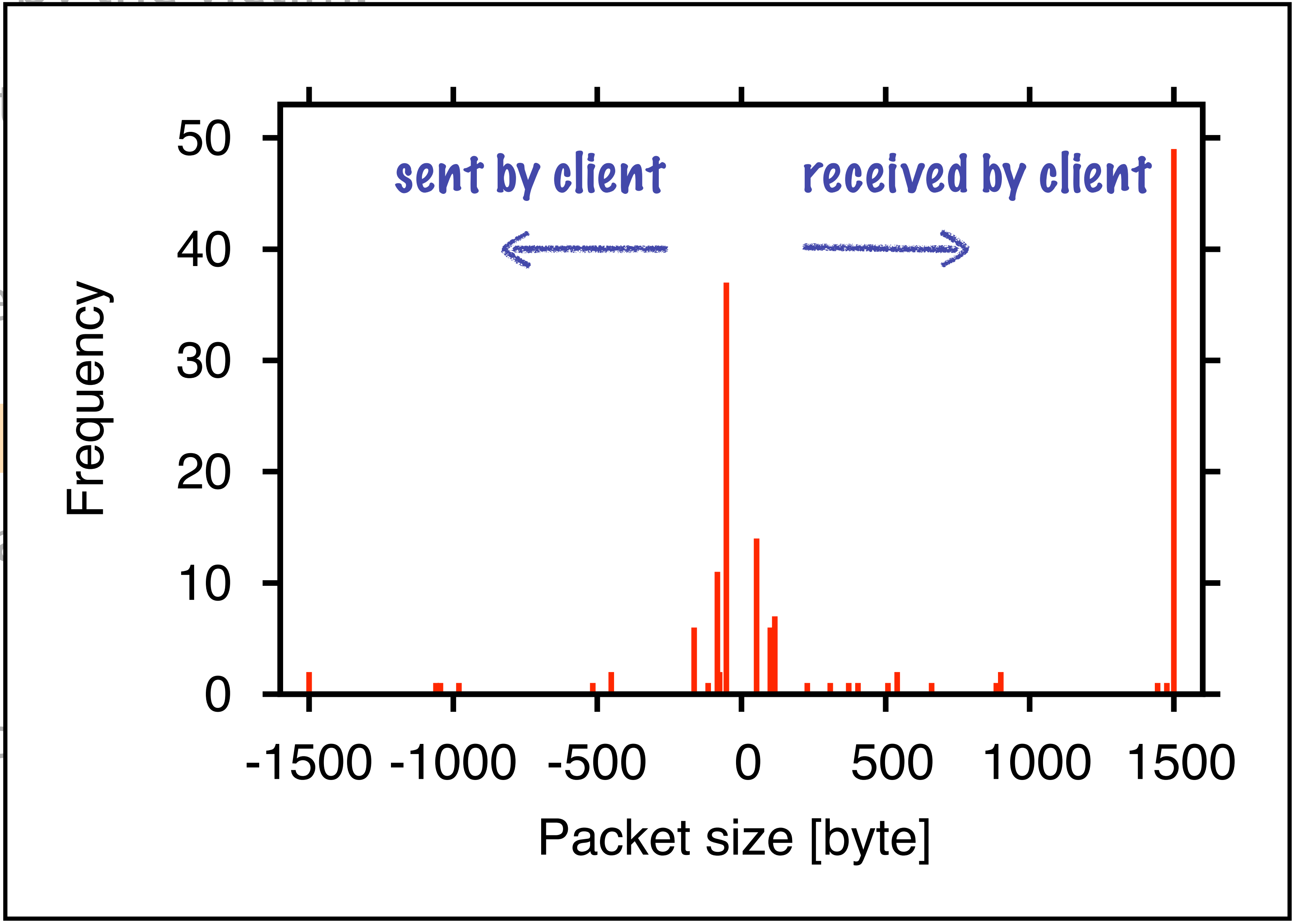
- Set up a database with traffic profiles of all websites of interest (training phase)
- Compare observed traffic with all profiles from database to predict likely candidates

# Overview of Our Fingerprinting Attack

- Attacker wants to learn URLs of websites that are requested over an encrypted tunnel by the victim.
- **Website Fingerprints:** At
of websites.
- **Attacker:** passive, local, e

**PROCEDURE**

- Set up a database with tra
(training phase)
- Compare observed traffic
predict likely candidates



# Overview of Our Fingerprinting Attack

- Attacker wants to learn URLs of websites that are requested over an encrypted tunnel by the victim.
- **Website Fingerprints:** Attack exploits characteristic structure of websites.
- **Attacker:** passive, local, external observer

Most PETs are supposed to protect against such harmless attackers!

**PROCEDURE**

- Set up a database with traffic profiles of all websites of interest (training phase)
- Compare observed traffic with all profiles from database to predict likely candidates

# Previous works concentrate on OpenSSH and two well-known fingerprinting techniques

Operating on file sizes:

- Sun et al. (2002)

**but: file sizes cannot be observed in encrypted tunnels!**

Operating on IP packet sizes:

- Bissias et al. (2005): **identify only 20% of sites**
- Liberatore & Levine (2006): identify up to 73% of sites
  **using Jaccard coefficient and Naïve-Bayes classifier**

# Focus of Our Paper

Operating on file sizes:

- Sun et al. (200

***Can we improve accuracy?***

**but: file sizes cannot be observed in encrypted tunnels!**

***What about other PETs?***

Operating on IP packet sizes:

- Bissias et al. (2005): **identify only 20% of sites**

***Does it work in practice?***

- Liberatore & L
  **using Jaccard coefficient and Naïve-Bayes classifier**

# Agenda

Motivation and Scenario

**Novel Fingerprinting Technique**

Evaluation

Addressing Real-World Issues

# Modeling Website Fingerprinting as Supervised Learning Problem

| | | |
|---|---|---|
| class | = | URLs |
| instance | = | observed IP packets |
| attribute | = | packet size |
| attribute value | = | packet size frequency |

**Example:**

- **class:** www.yahoo.com
- **some instance:** -160, 1500, 468, -52, 1500, 1500, -52, 1500
- **set representation:** (-160, -52, 468, 1500)
- **vector representation:** (1, 2, 1, 4)

# Review of Existing Fingerprinting Techniques

- **Jaccard Coefficient**
  - $sim(A, B) = |A \cap B| \;/\; (A \cup B)$; $sim(A, B) \in [0;1]$
  - Operates on set representation of instances
  - Poor accuracy for padded packets
- **Naïve Bayes Classifier**
  - Estimates probability density function for each packet size
  - Increased accuracy with *Kernel Density Estimation (KDE)*
  - Overfitting if only similar training instances are available

# Our Fingerprinting Technique: Multinomial Naïve Bayes (MNB) Classifier

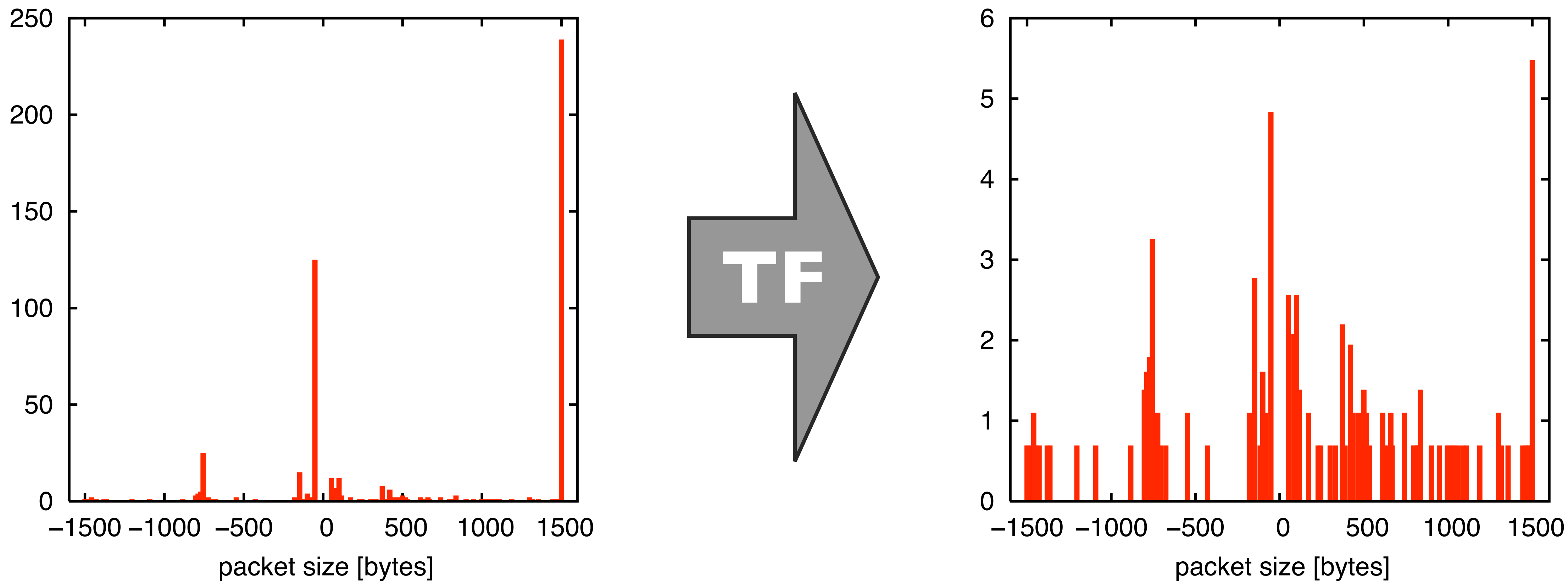
- Popular classifier in **text mining** domain (spam detection)
- We believe that Website Fingerprinting is a similar problem.

- Operates on packet size frequency distribution
- **Idea:** the more often the most important packet sizes of the test instance *i* appear in traces belonging to class *c*, the more likely does instance *i* belong to class *c*
- Low computational complexity

# Our Fingerprinting Technique: Transformations to Consider

Several optimisations to transform frequency vectors:

- **TF transformation**
  scale frequencies logarithmically to avoid bias towards classes with many packets with high frequencies



# Our Fingerprinting Technique: Transformations to Consider

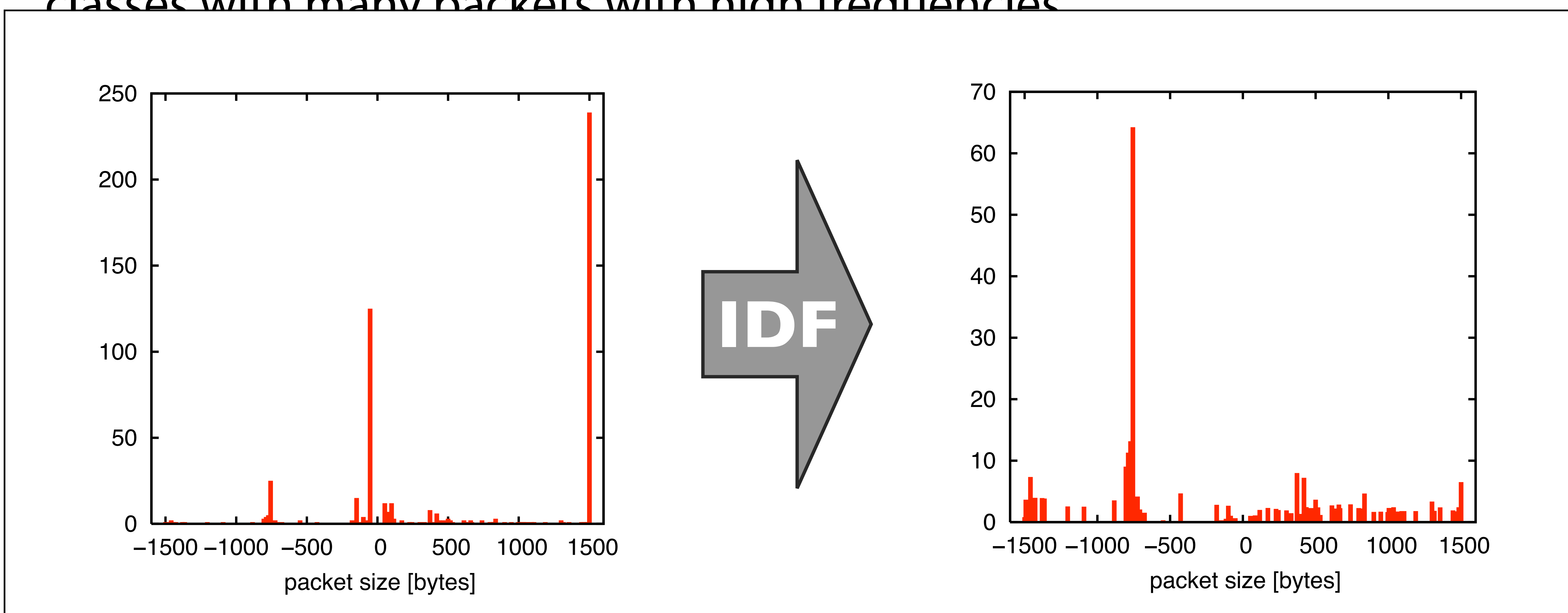
Several optimisations to transform frequency vectors:

- **TF transformation**
  scale frequencies logarithmically to avoid bias towards classes with many packets with high frequencies
- **IDF transformation**
  scale down frequencies of terms that are not characteristic for a class (inverse document frequency)

# Our Fingerprinting Technique: Transformations to Consider

Several optimisations to transform frequency vectors:

- **TF transformation**
  scale frequencies logarithmically to avoid bias towards classes with many packets with high frequencies



# Our Fingerprinting Technique: Transformations to Consider

Several optimisations to transform frequency vectors:

- **TF transformation**
  scale frequencies logarithmically to avoid bias towards classes with many packets with high frequencies
- **IDF transformation**
  scale down frequencies of terms that are not characteristic for a class (inverse document frequency)
- **Cosine normalisation**
  normalise attribute vectors to uniform length (division by Euclidean length of each vector)

# Agenda

- Motivation and Scenario
- Novel Fingerprinting Technique
- **Evaluation**
- Addressing Real-World Issues

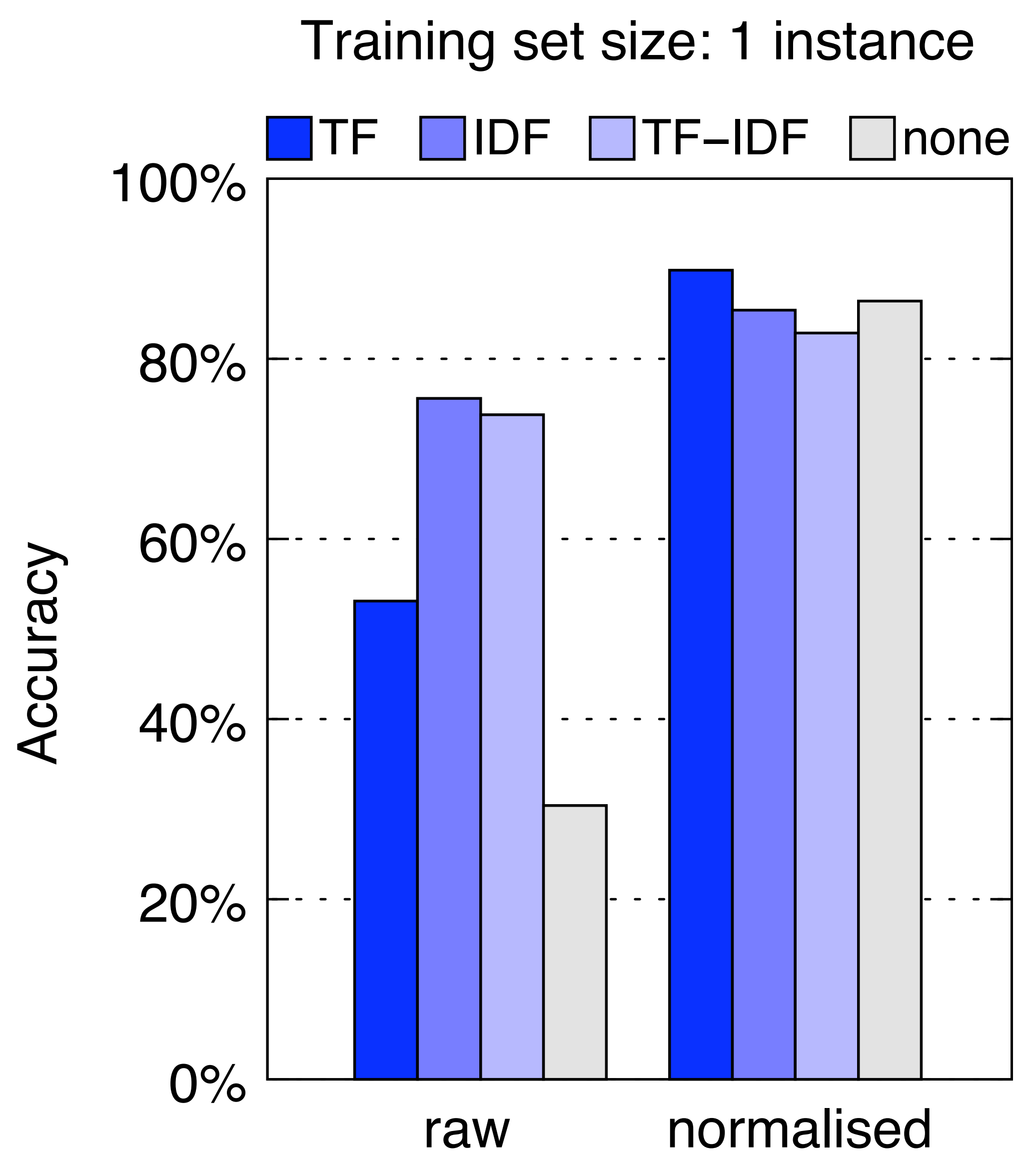
# Data Collection Methodology

- We obtained real-world traffic dumps from 775 popular domains
- Automated Firefox to download each site multiple times
- Recorded packet size and direction with *tcpdump*
- 300,000 traffic dumps for various PET systems within two months

Dataset will be available at our site for future research:
http://www-sec.uni-r.de/website-fingerprinting/

# Best Accuracy for TF Transformation and Normalisation

Normalisation makes classifier operate on relative packet frequencies



# More Results for OpenSSH

Multinomial Naïve Bayes with *TF* and *normalisation:*

- Already 90% accuracy for 1 training instance;  94% for 4 instances
- No substantial increase for more than 4 training instances

- Fingerprints built from frequency distribution of IP packet sizes are **very robust against changes to contents** of sites.
- Accuracy with old fingerprints decreases rather slowly: still over 90% after 17 days

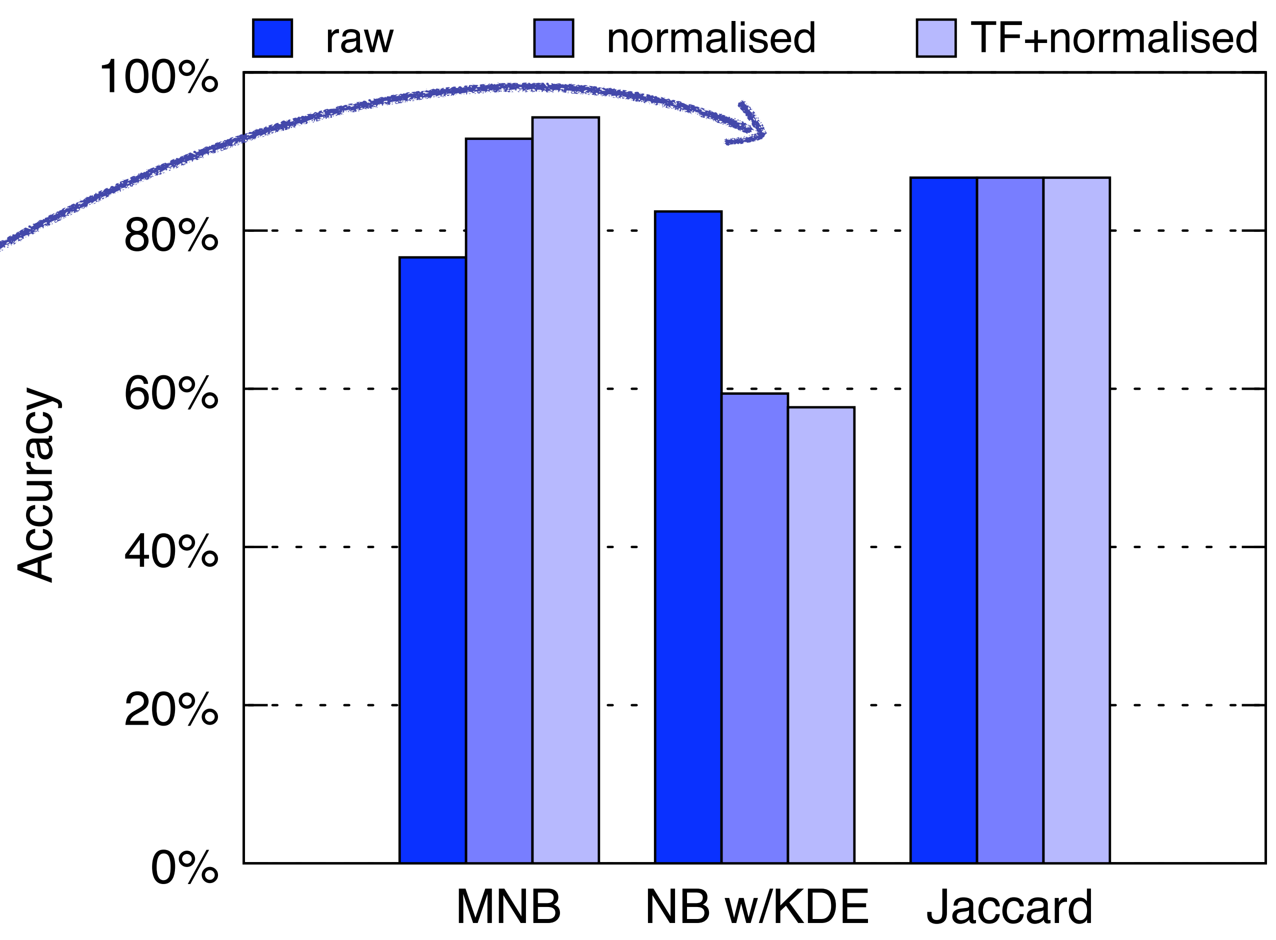
Cannot directly compare these results with previous work!

# Benchmarking Existing Website Fingerprinting Techniques with Our Sample

OpenSSH, 4 training and 4 test instances, *delta_t* = 6 days

- **highest accuracy:** MNB with TF+normalisation
- **Naïve Bayes** *really needs* absolute packet frequencies
- can reproduce good accuracy of **Jaccard coefficient** from previous work

NB with KDE and Jaccard perform better than in previous studies; i.e. results not comparable across samples!



# Attacking Popular PETs Using the MNB Classifier

## SINGLE HOP SYSTEMS

- Stunnel
- OpenSSH
- Cisco IPSec VPN
- OpenVPN

## MULTI HOP SYSTEMS

- JonDonym (*aka* JAP/AN.ON)
- Tor

# Attacking Popular PETs Using the MNB Classifier

| SINGLE HOP SYSTEMS | ACCURACY |
|---|---|
| Stunnel | 97.6% |
| OpenSSH | 96.7% |
| Cisco IPSec VPN | 96.2% |
| OpenVPN | 94.9% |

| MULTI HOP SYSTEMS | |
|---|---|
| JonDonym (*aka* JAP/AN.ON) | 20.0% |
| Tor | 3.0% |

# Attacking Popular PETs Using the MNB Classifier

| SINGLE HOP SYSTEMS | ACCURACY |
|---|---|
| Stunnel | 97.6% |
| OpenSSH | 96.7% |
| Cisco IPSec VPN | 96.2% |
| OpenVPN | 94.9% |

| MULTI HOP SYSTEMS | |
|---|---|
| JonDonym (*aka* JAP/AN.ON) | 20.0% |
| Tor | 3.0% |

Still way better than random guessing; p = 1 / 775 = 0.58%

# Attacking Popular PETs Using the MNB Classifier

| SINGLE HOP SYSTEMS | ACCURACY |
|---|---|
| Stunnel | 97.6% |
| OpenSSH | 96.7% |
| Cisco IPSec VPN | 96.2% |
| OpenVPN | 94.9% |

| MULTI HOP SYSTEMS | | with 10 guesses |
|---|---|---|
| JonDonym (*aka* JAP/AN.ON) | 20.0% | 47.5% |
| Tor | 3.0% | 22.1% |

# Attacking Popular PETs Using the MNB Classifier

| SINGLE HOP SYSTEMS | ACCURACY | | BEST CLASSIFIER |
|---|---|---|---|
| Stunnel | 97.6% | | TF-N |
| OpenSSH | 96.7% | | TF-N |
| Cisco IPSec VPN | 96.2% | | TF-N |
| OpenVPN | 94.9% | | TF-N |

| MULTI HOP SYSTEMS | | with 10 guesses | |
|---|---|---|---|
| JonDonym (*aka* JAP/AN.ON) | 20.0% | 47.5% | N |
| Tor | 3.0% | 22.1% | N |

# Attacking Popular PETs Using the MNB Classifier

| SINGLE HOP SYSTEMS | ACCURACY | BEST CLASSIFIER | NO. OF UNIQUE PACKET SIZES |
|---|---|---|---|
| Stunnel | 97.6% | TF-N | 1605 |
| OpenSSH | 96.7% | TF-N | 420 |
| Cisco IPSec VPN | 96.2% | TF-N | 108 |
| OpenVPN | 94.9% | TF-N | 2898 |

No correlation with accuracy!

with 10 guesses

| MULTI HOP SYSTEMS | ACCURACY | ACCURACY (with 10 guesses) | BEST CLASSIFIER | NO. OF UNIQUE PACKET SIZES |
|---|---|---|---|---|
| JonDonym (*aka* JAP/AN.ON) | 20.0% | 47.5% | N | 205 |
| Tor | 3.0% | 22.1% | N | 869 |

# Discussion of Results

- **OpenSSH results indicative** for all studied single-hop systems
- Low accuracies for multi-hop systems due to
  - **fixed-length packages** (e.g. Tor has cell size of 512 bytes)
  - **noise** (e.g. due to TCP retransmissions)

- We **cannot conclude** that multi-hop systems are immune against fingerprinting attacks!
- **System-specific attacks** will likely achieve higher accuracies.

# Agenda

Motivation and Scenario

Novel Fingerprinting Technique

Evaluation

**Addressing Real-World Issues**

# Research Assumptions

Results obtained using research assumptions from related studies:

- **Knowledge about victim:** attacker uses similar browser, Internet access and PET system to build fingerprints database
- **Closed-world:** classifier will never encounter traffic of a site it hasn‘t been trained for
- **Browser configuration:** no caching, no prefetching, no update checks
- **Extractable profiles:** attacker can extract traffic of individual page impressions from encrypted stream

# Evaluation of Two Real-World Issues with OpenSSH Dataset

## ENABLING BROWSER CACHE

- Previous work suggests that fingerprinting becomes difficult once **browser cache** is enabled.
- Cannot reproduce this with our sample: accuracy drops by only 5%

## FALSE ALARMS

- Leaving **closed world scenario** behind:
  false alarms for uninteresting sites become a problem
- If only 78 of 775 pages are considered *interesting*,
  - 1.5% of *uninteresting* instances cause a false alarm
  - 40% of instances from *interesting* sites are classified correctly

# Areas of Future Work

- Assess utility for **forensics**:
  tune attack for recognition of a very small number of sites
- Evaluate protection of **countermeasures**:
  e.g. *Traffic Flow Confidentiality* by Kiraly et al. (2008)
- Applicability to **Cloud Computing** protocols:
  must pay attention to traffic profile of messages

# Website Fingerprinting

- Introduced **Multinomial Naïve Bayes classifier**
- Operates on **transformed relative IP packet size frequencies**
- **Higher effectivity/efficiency** for OpenSSH than existing fingerprinting techniques (accuracy of up to 97%)
- Attack also relevant for **PETs with fixed-size messages** (with limited accuracy)
- **Browser caching** is apparently negligible


**Dominik Herrmann**, Hannes Federrath
University of Regensburg, Germany

Rolf Wendolsky
JonDos GmbH


Management of Information Security (Prof. Dr. Hannes Federrath)
http://www-sec.uni-r.de/website-fingerprinting/