

# Reinventing the Wheel: Publishing High-quality Slides

利用公共网关的 **SMS** 生态系统的安全性描述

答辩人: 李易峰

指导老师: 吴亦凡教授

中北大学英雄与联盟工程学院

February 15, 2018

# 目录

## 1 引言

- 研究背景
- 主要工作

## 2 System Design

- System Architecture
- Demo

## 3 Recap

- Ongoing Study
- 研究方法与数据集特征
- 算法
- Future Work

# 目录

## 1 引言

- 研究背景
- 主要工作

## 2 System Design

- System Architecture
- Demo

## 3 Recap

- Ongoing Study
- 研究方法与数据集特征
- 算法
- Future Work

# 研究背景



Figure: SUSTech Campus

- 短信息 (SMS) 成为现代通讯的重要组成部分
  - 很多组织或网站使用短信息作为身份验证的辅助通道

# 研究背景



Figure: SUSTech Campus

- 短信息 (SMS) 成为现代通讯的重要组成部分
  - 很多组织或网站使用短信息作为身份验证的辅助通道
- 现代短消息的发送，在抵达终端之前不接触蜂窝网络
  - 短信息 (SMS) 成为现代通讯的重要组成部分

# 主要工作

完成这项工作需要如下步骤

- 对 SMS 数据进行迄今为止最大的挖掘分析

# 主要工作

完成这项工作需要如下步骤

- 对 SMS 数据进行迄今为止最大的挖掘分析
- 评估良性短消息服务的安全态势

# 主要工作

完成这项工作需要如下步骤

- 对 SMS 数据进行迄今为止最大的挖掘分析
- 评估良性短消息服务的安全态势
- 刻画通过 SMS 网关进行的恶意行为



# OTT 服务



Figure: OTT 服务

OTT 服务支持在数据网络上提供短信和语音等第三方服务。  
OTT 可以使用云服务来存储和同步 SMS 到用户的其他设备。

# State of the art

- Current anti-procrastination systems lack raw force

# State of the art

- Current anti-procrastination systems lack raw force
  - Pomodoro, GTD, etc.

# State of the art

- Current anti-procrastination systems lack raw force
  - Pomodoro, GTD, etc.
- Raw force are often misused and in the wrong hands

# State of the art

- Current anti-procrastination systems lack raw force
  - Pomodoro, GTD, etc.
- Raw force are often misused and in the wrong hands
- People usually try to avoid punishment by telling lies

# 目录

## 1 引言

- 研究背景
- 主要工作

## 2 System Design

- System Architecture
- Demo

## 3 Recap

- Ongoing Study
- 研究方法与数据集特征
- 算法
- Future Work

# System architecture

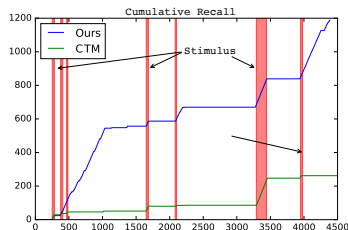


Figure: System components

# System architecture

## ■ Electric Chair: Punishment

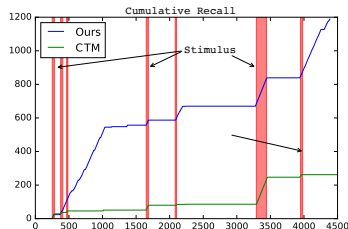


Figure: System components



# System architecture

- Electric Chair: Punishment
- Sensor: Detect behavior

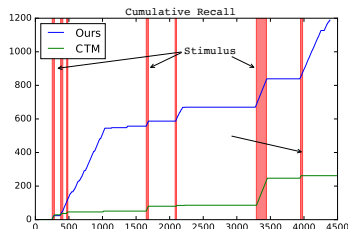


Figure: System components

# System architecture

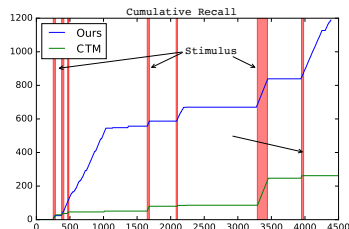


Figure: System components

- Electric Chair: Punishment
- Sensor: Detect behavior
- Handcuffs: Raw confinement

# Tracking Results

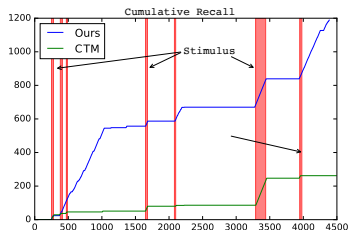
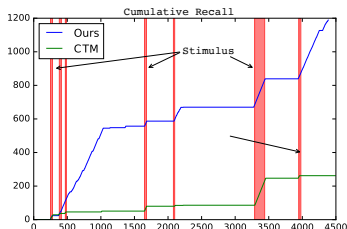


Figure: Effect of Electric Shock

# Tracking Results



- Electric shock indeed improves the memory of subjects

Figure: Effect of Electric Shock

# Tracking Results

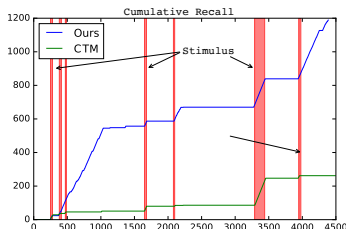


Figure: Effect of Electric Shock

- Electric shock indeed improves the memory of subjects
- This is a big loss for Big Brother

# Demo

# 目录

## 1 引言

- 研究背景
- 主要工作

## 2 System Design

- System Architecture
- Demo

## 3 Recap

- Ongoing Study
- 研究方法与数据集特征
- 算法
- Future Work

# Recap

In the paper:

- Proposed an electric-shock-based memory enhancement system that



# Recap

In the paper:

- Proposed an electric-shock-based memory enhancement system that
  - Uses handcuffs to confine user

# Recap

In the paper:

- Proposed an electric-shock-based memory enhancement system that
  - Uses handcuffs to confine user
  - Can maintain reliable performance in real-world conditions

# Recap

In the paper:

- Proposed an electric-shock-based memory enhancement system that
  - Uses handcuffs to confine user
  - Can maintain reliable performance in real-world conditions
- Implemented such a system and proved that

# Recap

In the paper:

- Proposed an electric-shock-based memory enhancement system that
  - Uses handcuffs to confine user
  - Can maintain reliable performance in real-world conditions
- Implemented such a system and proved that
  - It really works™

# 研究方法 with 数据集特征

Table: 公共网关及抓取的信息数

Site	Message
receivesmsonline.net	81313
receive-sms-online.info	69389
receive-sms-now.com	63797
hs3x.com	55499
receivesmsonline.com	44640
receivefreesms.com	37485
receive-sms-online.com	27094
e-receivesms.com	7107

# 研究方法与数据集特征

## ■ 使用 Scrapy 框架爬取公共网关

Table: 公共网关及抓取的信息数

Site	Message
receivesmsonline.net	81313
receive-sms-online.info	69389
receive-sms-now.com	63797
hs3x.com	55499
receivesmsonline.com	44640
receivefreesms.com	37485
receive-sms-online.com	27094
e-receivesms.com	7107

# 研究方法 with 数据集特征

- 使用 Scrapy 框架爬取公共网关
- 收集 8 个公共短信网关在 14 个月的数据

**Table:** 公共网关及抓取的信息数

Site	Message
receivesmsonline.net	81313
receive-sms-online.info	69389
receive-sms-now.com	63797
hs3x.com	55499
receivesmsonline.com	44640
receivefreesms.com	37485
receive-sms-online.com	27094
e-receivesms.com	7107

# 研究方法 with 数据集特征

- 使用 Scrapy 框架爬取公共网关
- 收集 8 个公共短信网关在 14 个月的数据
- 共抓取 386,327 条数据

Table: 公共网关及抓取的信息数

Site	Message
receivesmsonline.net	81313
receive-sms-online.info	69389
receive-sms-now.com	63797
hs3x.com	55499
receivesmsonline.com	44640
receivefreesms.com	37485
receive-sms-online.com	27094
e-receivesms.com	7107



# 消息聚类分析

## 基本思路

# 消息聚类分析

## 基本思路

- 使用编辑距离矩阵将类似的消息归于一张连通图中。

# 消息聚类分析

## 基本思路

- 使用编辑距离矩阵将类似的消息归于一张连通图中。
- 使用固定值替换感兴趣的消息，如代码、email 地址。

# 消息聚类分析

## 基本思路

- 使用编辑距离矩阵将类似的消息归于一张连通图中。
- 使用固定值替换感兴趣的消息，如代码、email 地址。
- 查找归一化距离小于阈值的消息，并确定聚类边界。

# 消息聚类分析

## 基本思路

- 使用编辑距离矩阵将类似的消息归于一张连通图中。
- 使用固定值替换感兴趣的消息，如代码、email 地址。
- 查找归一化距离小于阈值的消息，并确定聚类边界。

## 实现步骤

# 消息聚类分析

## 基本思路

- 使用编辑距离矩阵将类似的消息归于一张连通图中。
- 使用固定值替换感兴趣的消息，如代码、email 地址。
- 查找归一化距离小于阈值的消息，并确定聚类边界。

## 实现步骤

- 1 加载所有消息。

# 消息聚类分析

## 基本思路

- 使用编辑距离矩阵将类似的消息归于一张连通图中。
- 使用固定值替换感兴趣的消息，如代码、email 地址。
- 查找归一化距离小于阈值的消息，并确定聚类边界。

## 实现步骤

- 1 加载所有消息。
- 2 用固定的字符串替换数字、电子邮件和 URL 以预处理消息。

# 消息聚类分析

## 基本思路

- 使用编辑距离矩阵将类似的消息归于一张连通图中。
- 使用固定值替换感兴趣的消息，如代码、email 地址。
- 查找归一化距离小于阈值的消息，并确定聚类边界。

## 实现步骤

- 1 加载所有消息。
- 2 用固定的字符串替换数字、电子邮件和 URL 以预处理消息。
- 3 将预处理后的信息按字母排序。



# 消息聚类分析

## 基本思路

- 使用编辑距离矩阵将类似的消息归于一张连通图中。
- 使用固定值替换感兴趣的消息，如代码、email 地址。
- 查找归一化距离小于阈值的消息，并确定聚类边界。

## 实现步骤

- 1 加载所有消息。
- 2 用固定的字符串替换数字、电子邮件和 URL 以预处理消息。
- 3 将预处理后的信息按字母排序。
- 4 通过使用编辑距离阈值 (0.9) 来确定聚类边界。

# 消息聚类分析

## 基本思路

- 使用编辑距离矩阵将类似的消息归于一张连通图中。
- 使用固定值替换感兴趣的消息，如代码、email 地址。
- 查找归一化距离小于阈值的消息，并确定聚类边界。

## 实现步骤

- 1 加载所有消息。
- 2 用固定的字符串替换数字、电子邮件和 URL 以预处理消息。
- 3 将预处理后的信息按字母排序。
- 4 通过使用编辑距离阈值 (0.9) 来确定聚类边界。
- 5 手动标记各个聚类，以确定服务提供者、消息类别等。

# 算法

---

算法 1 algorithm caption

---

**Input:** input parameters A, B, C

**Output:** output result

# 算法

---

## 算法 2 algorithm caption

---

**Input:** input parameters A, B, C

**Output:** output result

1: some description

# 算法

---

## 算法 3 algorithm caption

---

**Input:** input parameters A, B, C

**Output:** output result

- 1: some description
- 2: condition

# 算法

---

## 算法 4 algorithm caption

---

**Input:** input parameters A, B, C

**Output:** output result

- 1: some description
- 2: condition
- 3: ...

# 算法

---

## 算法 5 algorithm caption

---

**Input:** input parameters A, B, C

**Output:** output result

- 1: some description
- 2: condition
- 3: ...
- 4: condition

# 算法

---

## 算法 6 algorithm caption

---

**Input:** input parameters A, B, C

**Output:** output result

- 1: some description
- 2: condition
- 3: ...
- 4: condition
- 5:



# 算法

---

## 算法 7 algorithm caption

---

**Input:** input parameters A, B, C

**Output:** output result

- 1: some description
- 2: condition
- 3: ...
- 4: condition
- 5:
- 6:

# 算法

---

## 算法 8 algorithm caption

---

**Input:** input parameters A, B, C

**Output:** output result

1: some description

2: condition

3: ...

4: condition

5:

6:

7:

# 算法

---

## 算法 9 algorithm caption

---

**Input:** input parameters A, B, C

**Output:** output result

- 1: some description
- 2: condition
- 3: ...
- 4: condition
- 5:
- 6:
- 7:
- 8: condition

# 算法

---

## 算法 10 algorithm caption

---

**Input:** input parameters A, B, C

**Output:** output result

1: some description

2: condition

3: ...

4: condition

5:

6:

7:

8: condition

9:

# 算法

---

## 算法 11 algorithm caption

---

**Input:** input parameters A, B, C

**Output:** output result

1: some description

2: condition

3: ...

4: condition

5:

6:

7:

8: condition

9:

10: **return** result

---

# 代码

```
1 inline int gcd(int a, int b) {  
2     return b==0?a:gcd(b,a%b)  
3 }  
4 inline int lcm(int a, int b) {  
5     return a/gcd(a,b)*b;  
6 }
```

# Future Work

- Get more people to try this

# Future Work

- Get more people to try this
- Benchmark the entire system in the wild



# Future Work

- Get more people to try this
- Benchmark the entire system in the wild
- Profit!

# Thank you

# Thank you for listening!

## Q&amp;A

# Questions?