Big Data, Little Data, No Such Data

Christian Grothoff

March 23, 2017

*"***Obedience** *is a direct form of social influence where an individual submits to, or complies with, an authority figure. Obedience may be explained by factors such as* **diffusion of responsibility**, *(...)*
Compliance can be achieved through various techniques (...).
Conversely, efforts to reduce obedience may be effectively based around **educating** *people (...) and exposing them to* **examples of disobedience**.*"*
—TOP SECRET JTRIG Report on Behavioural Science

Part I: Big Data[1]

[1] Joint work with Yves Eudes (FR), Monika Ermert (DE) and Jens Porup (EN)

NSA SKYNET

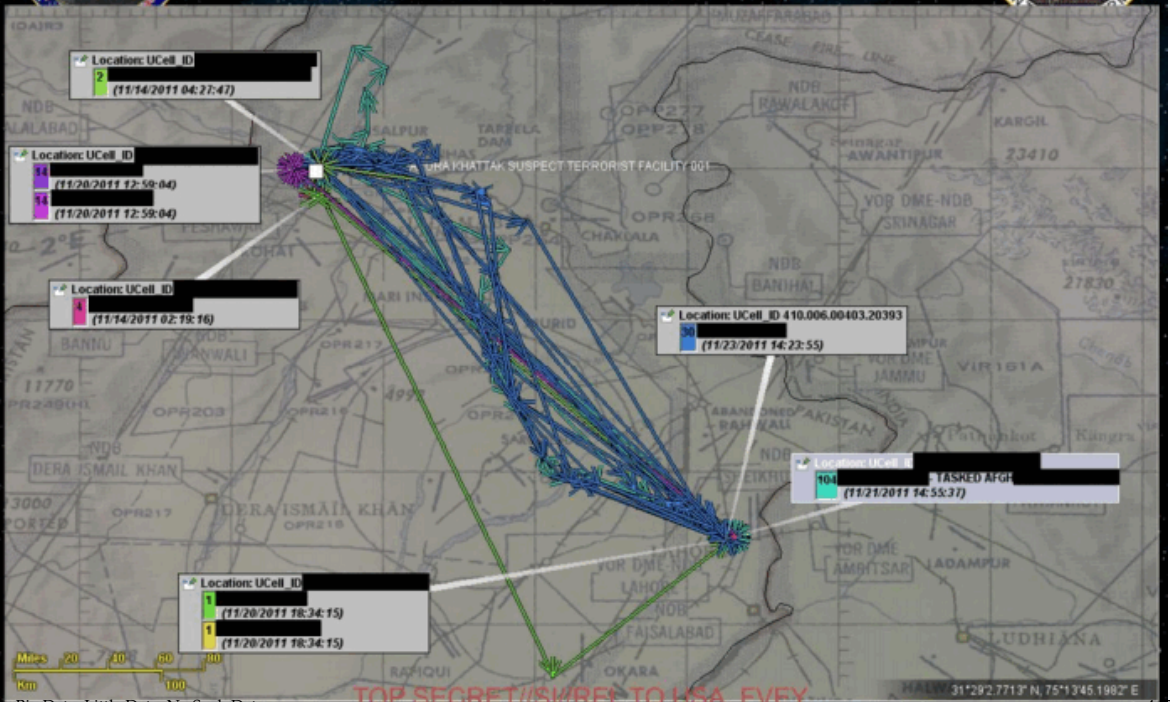

TOP SECRET//COMINT//REL TO USA, AUS, CAN, GBR, NZL
SKYNET: Applying Advanced Cloud-based Behavior Analytics
A Collaborative Project by S2I, R6, T12, T14, SSG, and S22
Presenters:
, S2I51
, R66F
Derived From: NSA/CSSM 1-52
Dated: 20070108
Declassify On: 20370401
TOP SECRET//COMINT//REL TO USA, AUS, CAN, GBR, NZL

Cloud Analytic Building Blocks

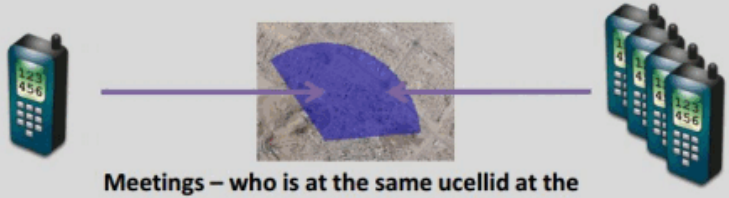
- Travel Patterns
  - Travel phrases (Locations visited in given timeframe)
  - Regular/repeated visits to locations of interest
- Behavior-Based Analytics
  - Low use, incoming calls only
  - Excessive SIM or Handset swapping
  - Frequent Detach/Power-down
  - Courier machine learning models
- Other Enrichments
  - Travel on particular days of the week
  - Co-travelers
  - Similar travel patterns
  - Common contacts
  - Visits to airports
  - Other countries
  - Overnight trips
  - Permanent move

SMARTTRACKER Travel View

31 October – 23 November



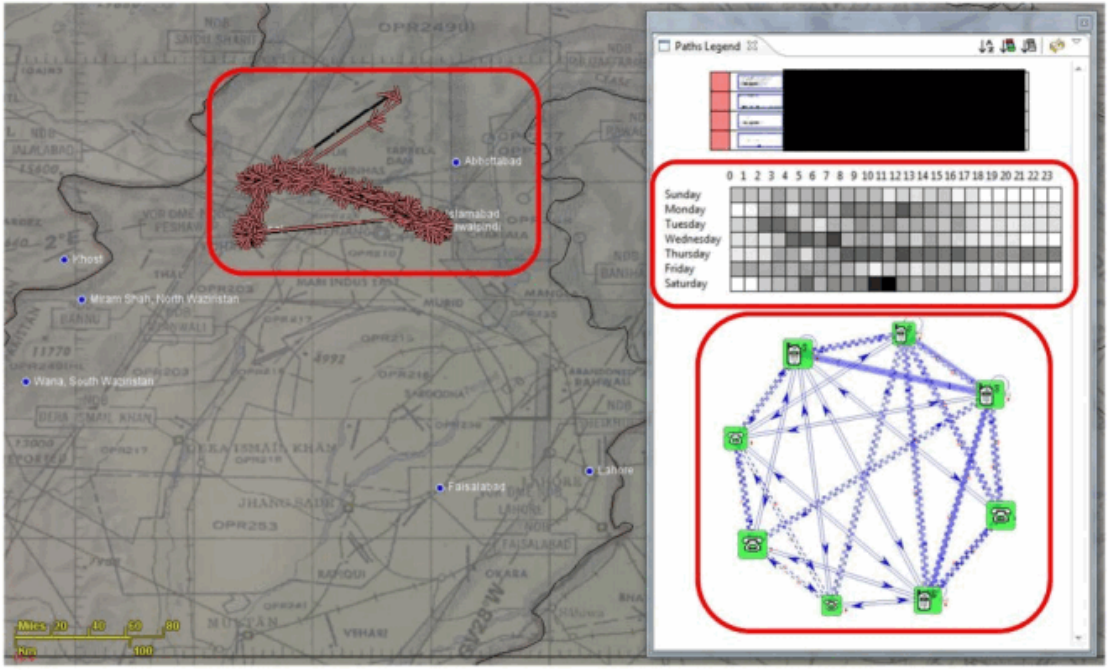
RT-RG Analytics



Meetings – who is at the same ucellid at the same time as the potential courier at the destination city?...Multiple times.

Sidekicks – is there a pair traveling together to the destination city?

From GSM metadata, we can measure aspects of each selector's pattern-of-life, social network, and travel behavior



TOP SECRET//COMINT//REL TO USA, FVEY

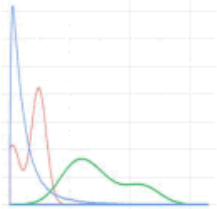
Analytic Tradecraft

- Examine travel patterns for common routes and meeting locations
  - Run cell soaks on all common meeting locations during meeting timeframe
- Analyze selectors for common contacts
- Analyze selectors for handset sharing behavior

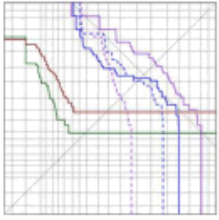
Repeat procedure with resulting selectors

Correlate with other known and suspected selectors

This presentation describes our search for AQSL couriers using behavioral profiling



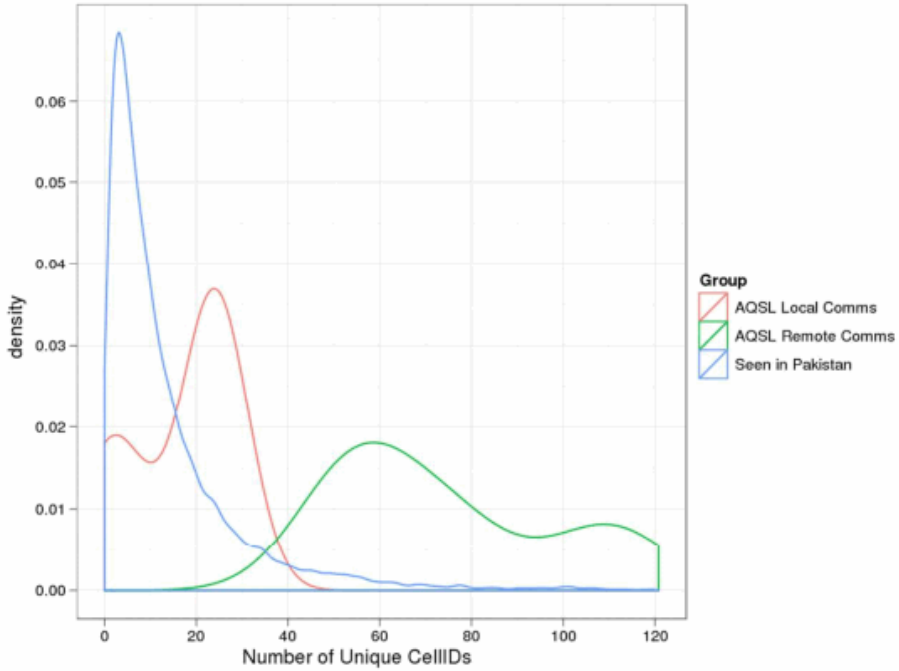
Behavioral Feature Extraction



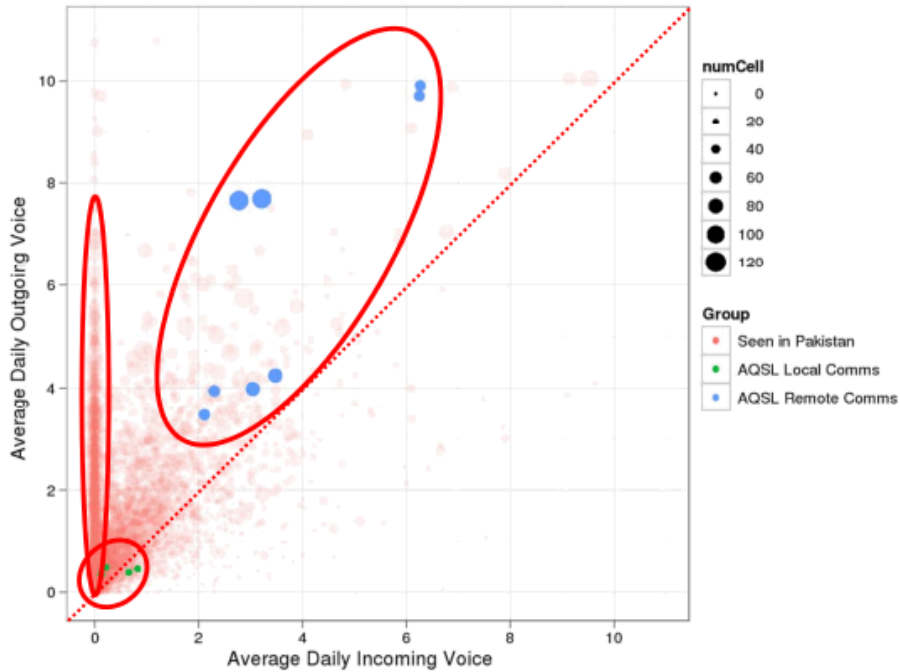
Cross Validation Experiment on AQSL Couriers

Preliminary SIGINT Findings

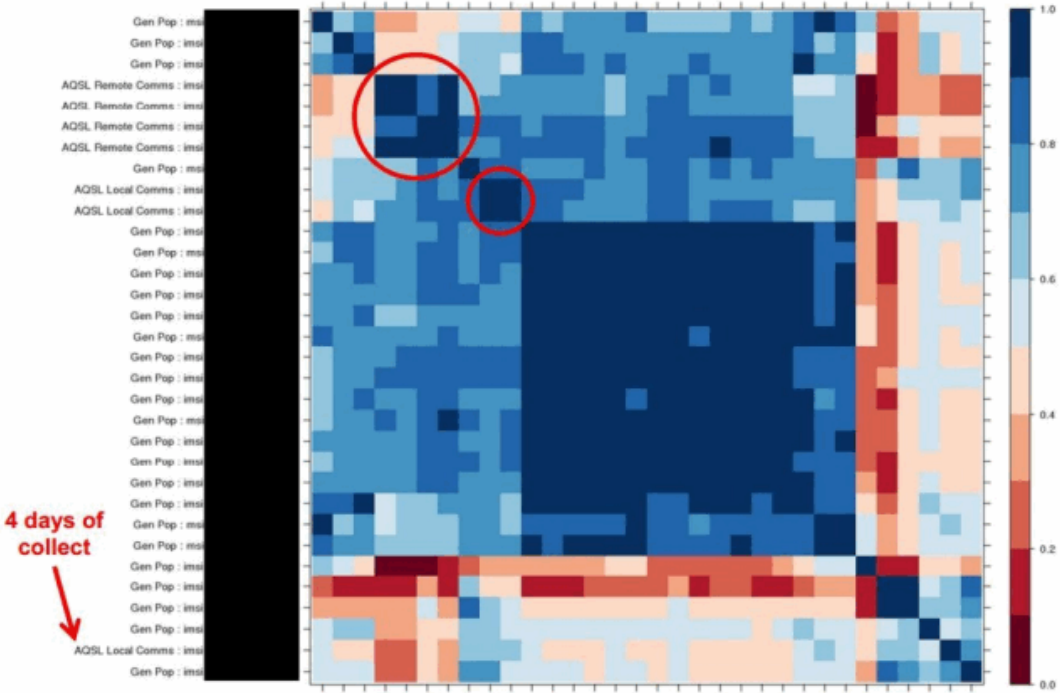
Counting unique UCELLIDs shows that couriers travel more often than typical Pakistani selectors



By examining multiple features at once, we can see some indicative behaviors of our courier selectors



Looking at a hierarchical clustering derived from all 80 features, the AQSL groups mostly stay together

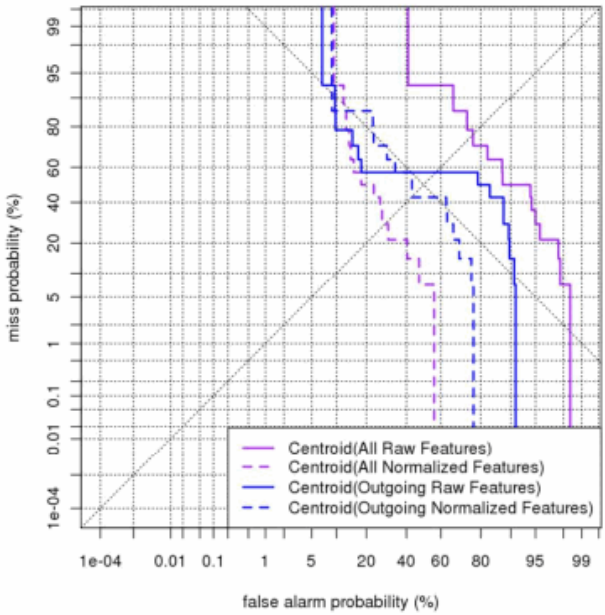


TOP SECRET//COMINT//REL TO USA, FVEY

Our initial detector uses the centroid of the AQSL couriers to "find other selectors like these"

AQSL Cross-Validation Experiment

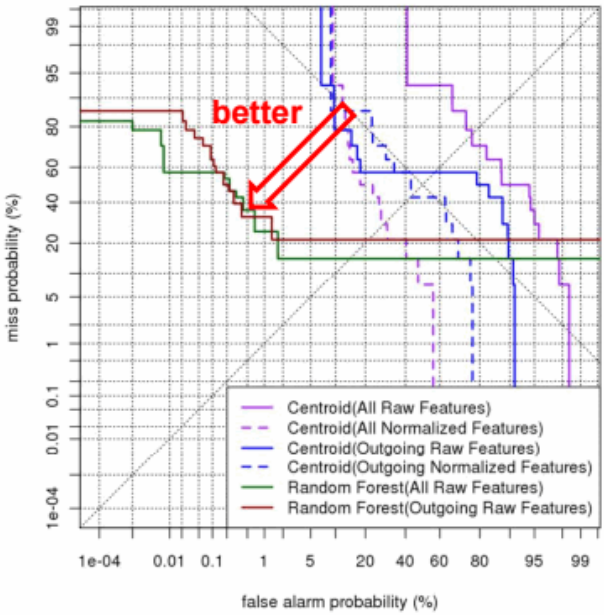
- 7 MSISDN/IMSI pairs
- Hold each pair out and score them when training the centroid on the rest
- Assume that random draws of Pakistani selectors are nontargets
- How well do we do?



Statistical algorithms are able to find the couriers at very low false alarm rates, if we're allowed to miss half of them

Random Forest Classifier

- 7 MSISDN/IMSI pairs
- Hold each pair out and then try to find them after learning how to distinguish remaining couriers fro n other Pakistanis
  (using 100k random selectors here)
- Assume that random draws of Pakistani selectors are nontargets
- 0.18% False Alarm Rate at 50% Miss Rate



Now, we'll investigate some findings after running these classifiers on +55M Pakistani selectors via MapReduce

Behavioral Feature Extraction

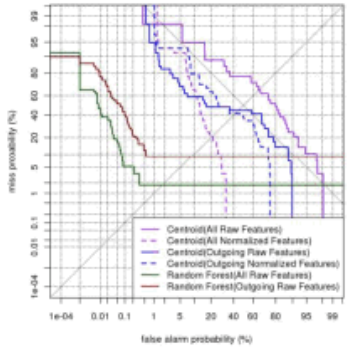
Cross Validation Experiment on AQSL Couriers

Preliminary SIGINT Findings

Preliminary results indicate that we're on the right track, but much remains to be done

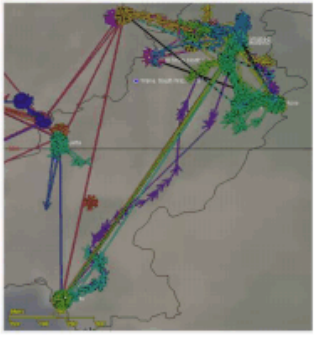
Cross Validation Experiment:

- Random Forest classifier operating at 0.18% false alarm rate at 50% miss
- Enhancing training data with Anchory selectors reduced that to 0.008%
- Mean Reciprocal Rank is ~1/10



Preliminary SIGINT Findings:

- Behavioral features helped discover similar selectors with "courier-like" travel patterns
- High number of tasked selectors at the top is hopefully indicative of the detector performing well "in the wild"



192 Million people live in Pakistan.

- 0.18% of the Pakistani population = 343,800 innocent citizens
- 0.008% of the Pakistani population = 15,280 innocent citizens

192 Million people live in Pakistan.

- 0.18% of the Pakistani population = 343,800 innocent citizens
- 0.008% of the Pakistani population = 15,280 innocent citizens

This is with half of AQSL couriers surviving the genocide.

> *"We kill based on metadata."*
>
> *—Michael Hayden (former NSA & CIA director)*

Further reading[2]

- Christian Grothoff and Yves Eudes. *Comment fonctionne Skynet, le programme ultra-secret de la NSA créé pour tuer*. **Le Monde**, 20.10.2015.
- Christian Grothoff and Monika Ermert. *Data Mining für den Drohnenkrieg*. **c't**, 3/2016.
- Christian Grothoff and Jens Porup. *The NSA's SKYNET program may be killing thousands of innocent people*. **ARS Technica**, 16.2.2016.
- Dave Gershgorn. *Can The NSA's Machines Recognzie a Terrorist?* **Popular Science**, 16.2.2016.
- Antonio Caffo. *NSA e quella tecnologia che non va oltre Facebook. Gli algoritmi utilizzati dalla National Security Agency in Pakistan dovrebbero identificare potenziali minacce. Ecco perché non ci riescono*, **Panorama.it**, 17.2.2016.
- Keskiviikko. *Ihmisoikeustutkija väittää: NSA:n SKYNET-algoritmi tappaa viattomia ihmisiä*, **Iltalehti.fi**, 17.2.2016.
- Martin Robbins. *Has a rapmaging AI algorithm really killed thousands in Pakistan?*, **The Guardian**, 18.2.2016.
- John Naughton. *Death by drone strike, dished out by algorithm*, **The Guardian**, 21.2.2016.

[2] RU, CN, JP references ommited due to rendering issues.

Part II: Little Data[3]

"Das ist das Geheimnis der Propaganda; den, den die Propaganda fassen will, ganz mit den Ideen der Propaganda zu durchtränken, ohne dass er überhaupt merkt, dass er durchtränkt wird."

—Joseph Goebbels

[3] Joint work with Álvaro García-Recuero and Jeffrey Burdges

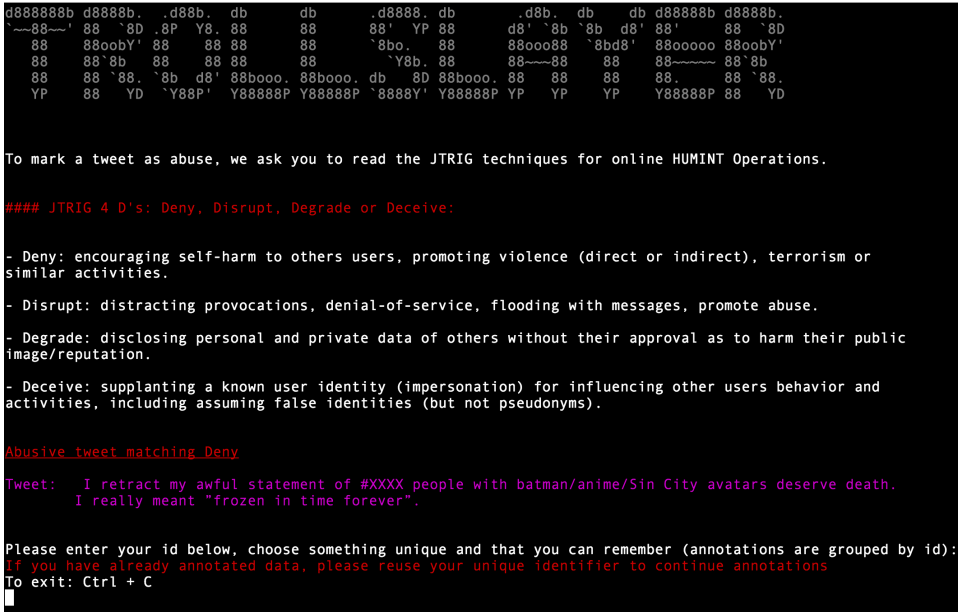
The Joint Threat Research and Intelligence Group (JTRIG)

> *2.3 (...) Generally, the language of JTRIG's operations is characterised by terms such as "discredit", promote "distrust", "dissuade", "deceive", "disrupt", "delay", "deny", "denigrate/degrade", and "deter".*

`http://www.statewatch.org/news/2015/jun/behavioural-science-support-for-jtrigs-effects.pdf`

Goal: Abuse detection in OSNs

Use machine learning to detect spam, fake accounts, or harassment in OSNs.



The Human Score

reviewer	total # reviewed	% abusive	% acceptable	# agreement	c-abusive	c-acceptable	c-overall
1	754	3.98	83.55	703	0.71	0.97	0.93
2	744	4.30	82.79	704	0.66	0.97	0.94
3	559	5.01	83.90	526	0.93	0.95	0.94
4	894	4.03	71.92	807	0.61	0.94	0.90
5	939	5.54	69.54	854	0.88	0.90	0.91
6	1003	5.68	69.79	875	0.95	0.89	0.87
average	816	4.76	76.92	**745**	**0.79**	**0.94**	**0.92**
std. dev.	162	0.76	7.18	130	0.15	0.03	0.03

Ground Zero: Twitter

Idea: Build "metadata-based" features by extracting information from a tweet, its author and social graph.

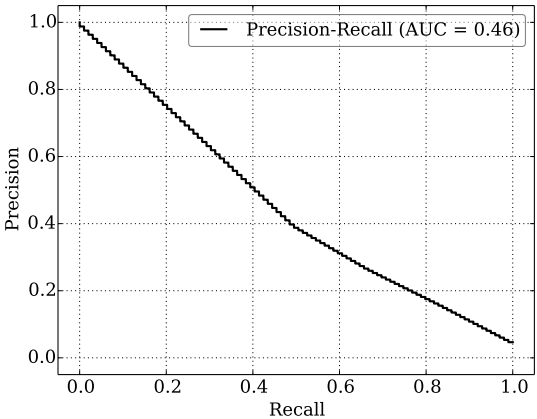
Examples:

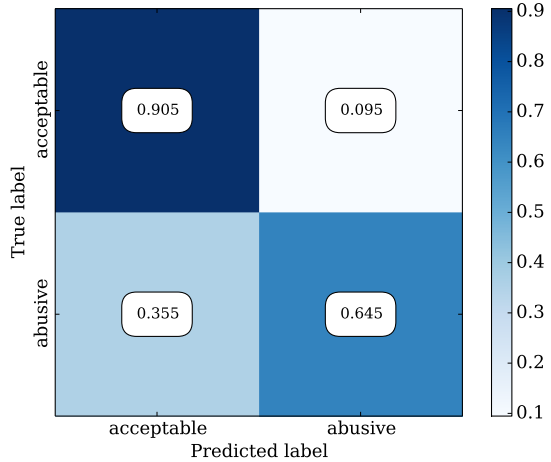
- Tweet invasive: do sender and receiver of tweet follow each other?
- Do sender and receiver share subscriptions?
- Account: how old is the account?

Features: The Long List

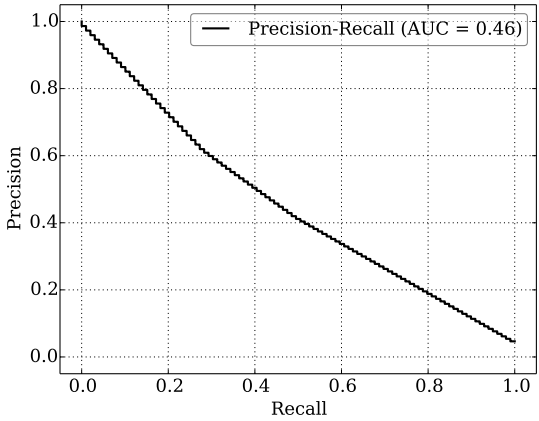
	Feature	Description
5.1	# lists	how many lists the sender has created
	# subscriptions	number of subscriptions of the sender
	$\frac{\#\text{ subscriptions}}{\text{age}}$	ratio of subscriptions made in relation to age of sender account
	$\frac{\#\text{subscriptions}}{\#\text{subscribers}}$	ratio of subscriptions to subscribers of sender
5.2	# mentions	number of mentions in the message
	# hashtags	number of hashtags in the message
5.3	message invasive	false if sender subscribed to receiver and receiver subscribed to sender
5.4	$\frac{\#\text{ messages}}{\text{age}}$	fraction of messages from sender in relation to its account age
	# retweets	number of retweets the sender has posted
	# favorited messages	number of messages favorited by sender
5.5	age of account	days since sender account creation
5.6	# subscribers	number of subscribers to public feed of the sender
	$\frac{\#\text{ subscribers}}{\text{age}}$	ratio of subscribers in relation to age of sender account
5.7	subscription $\cap$ subscription	size of the intersection among subscriptions of sender and receiver
5.8	subscriber $\cap$ subscriber	size of the intersection among subscribers of sender and receiver
5.9	$\text{subscriber}^r \cap \text{subscription}^s$	size of the intersection among subscribers of receiver and subscriptions of sender
	$\text{subscription}^r \cap \text{subscriber}^s$	size of the intersection among subscriptions of receiver and subscribers of sender

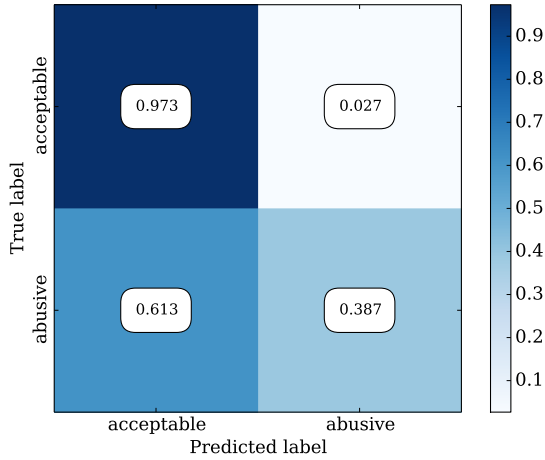
Extra Trees


Precision-Recall (AUC = 0.46)
Precision
Recall
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0


0.905
0.095
0.355
0.645
True label
acceptable
abusive
acceptable
abusive
Predicted label
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1

Gradient Boosting



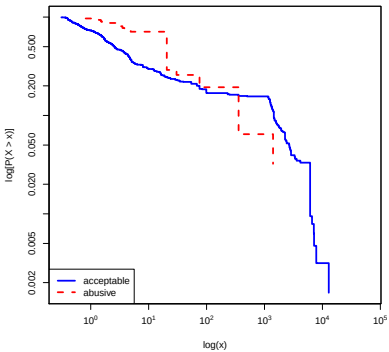


Thinking past Twitter

What about adversarial learning with privacy?

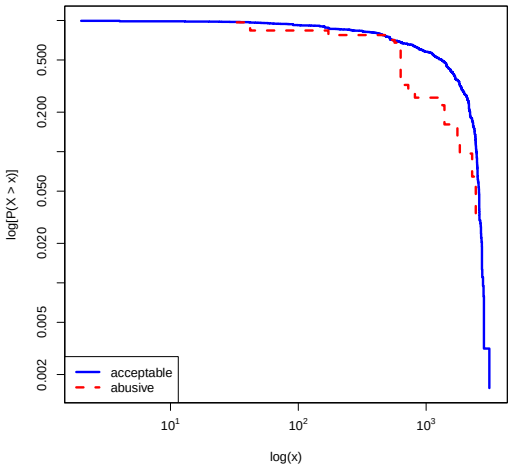
- Do not want to expose user metadata
- Do not want to expose activity metadata
- Do not want to expose social graph metadata

Detect Abuse



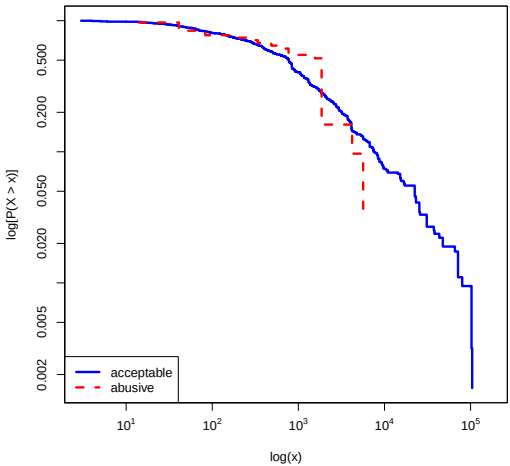
- (complementary CDF) CCDF of **messages per day**: how often is it (the random variable) above a particular level? No clear trend.
- Privacy? Seems OK for public messages.
- Security? Monitor via anonymous subscriptions to detect lying.

Detect Abuse



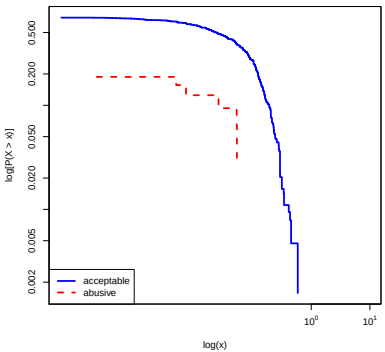
- CCDF shows **age of account** has a lower probability distribution for abusive accounts of older age.
- Privacy? Probably not an issue
- Security? Needs time-stamping service.

Detect Abuse



- CCDF of **number of subscribers** of the users shows no clear trend, presumably due to attackers artificially increasing their count.
- Privacy? Not huge issue.
- Security? Hard, proof-of-work may help a bit.

Detect Abuse



- CCDF of **Subscription** ∩ **Subscription** shows less overlap in subscriptions of the authors of abusive messages and subscriptions of the potential victims.
- Privacy? Protocol 1.
- Security? Hard to prevent fake accounts.

Straw-man version of protocol 1

Problem: Alice wants to compute $n := |\mathcal{L}_A \cap \mathcal{L}_B|$

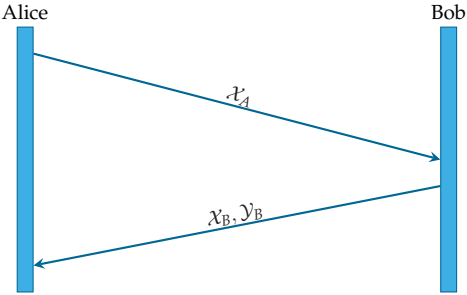
Suppose each user has a private key c_i and the corresponding public key is $C_i := g^{c_i}$ where g is the generator

The set up is as follows:

- $\mathcal{L}_A$: set of public keys representing Alice's subscriptions
- $\mathcal{L}_B$: set of public keys representing Bob's subscriptions
- Alice picks an ephemeral private scalar $t_A \in \mathbb{F}_p$
- Bob picks an ephemeral private scalar $t_B \in \mathbb{F}_p$

Straw-man version of protocol 1

$$\mathcal{X}_A := \{\, C^{t_A} \mid C \in \mathcal{L}_A \,\}$$



$$\mathcal{X}_B := \{\, C^{t_B} \mid C \in \mathcal{L}_B \,\}$$
$$\mathcal{Y}_B := \left\{\, \overline{C}^{t_B} \;\middle|\; \overline{C} \in \mathcal{X}_A \right\}$$
$$= \{\, C^{t_B \cdot t_A} \mid C \in \mathcal{L}_B \,\}$$

$$\mathcal{Y}_A := \left\{\, \hat{C}^{t_A} \;\middle|\; \hat{C} \in \mathcal{X}_B \right\}$$
$$= \{\, C^{t_A \cdot t_B} \mid C \in \mathcal{L}_A \,\}$$

Alice can get $|\mathcal{Y}_A \cap \mathcal{Y}_B|$ at linear cost.

Attacks against the Straw-man

If Bob controls two subscribers $C_1, C_2 \in \mathcal{L}_A$, he can:

- Detect relationship between $C_1^{t_A}$ and $C_2^{t_B}$
- Choose $K \subset \mathbb{F}_p$ and insert fakes:

$$\mathcal{X} := \bigcup_{k \in K} \left\{ C_1^k \right\}$$
$$\mathcal{Y} := \bigcup_{k \in K} \left\{ (C_1^{t_A})^k \right\}$$

so that Alice computes $n = |K|$.

Cut & choose version of protocol 1: Preliminaries

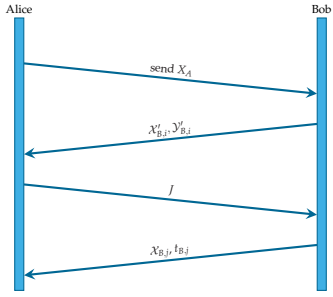
Assume a fixed system security parameter $\kappa \geq 1$.

Let Bob use secrets $t_{B,i}$ for $i \in \{1, \ldots, \kappa\}$, and let $\mathcal{X}_{B,i}$ and $\mathcal{Y}_{B,i}$ be blinded sets over the different $t_{B,i}$ as in the straw-man version.

For any list or set Z, define

$$Z' := \{h(x) | x \in Z\} \tag{1}$$

Cut & choose version of protocol 1



Protocol messages:

1. Alice sends:
   $\mathcal{X}_A := \texttt{sort}\left[C^{t_A} \;\middle|\; C \in \mathcal{A} \right]$
2. Bob responds with commitments:
   $\mathcal{X}'_{B,i}, \mathcal{Y}'_{B,i} \quad \textit{for} \;\; i \in 1, \ldots, \kappa$
3. Alice picks a non-empty random subset $J \subseteq \{1, \ldots, \kappa\}$ and sends it to Bob.
4. Bob replies with $\mathcal{X}_{B,j}$ for $j \in J$, and $t_{B,j}$ for $j \notin J$.

Cut & choose version of protocol 1: Verification

For $j \notin J$, Alice checks the $t_{B,j}$ matches the commitment $\mathcal{Y}'_{B,j}$.

For $j \in J$, she verifies the commitment to $\mathcal{X}_{B,j}$ and computes:

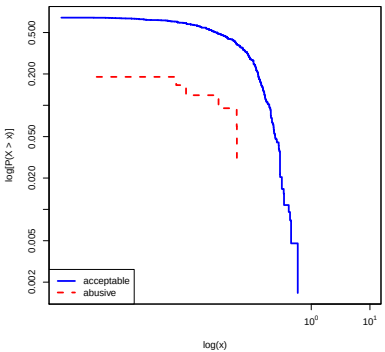
$$\mathcal{Y}_{A,j} := \left\{ \hat{C}^{t_A} \;\middle|\; \hat{C} \in \mathcal{X}_{B,j} \right\} \tag{2}$$

To get the result, Alice computes:

$$n = |\mathcal{Y}'_{A,j} \cap \mathcal{Y}'_{B,j}| \tag{3}$$

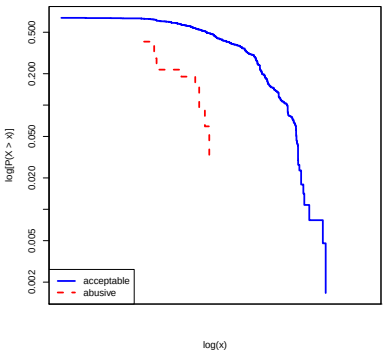
Alice checks that the n values for all $j \in J$ agree.

Detect Abuse



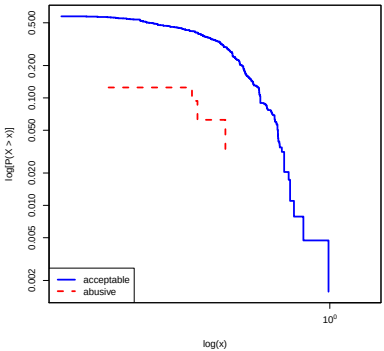
- CCDF of **Subscription** ∩ **Subscription** shows less overlap in subscriptions of the authors of abusive messages and subscriptions of the potential victims.
- Privacy? Protocol 1.
- Security? Hard to prevent fake accounts.

Privacy Analysis of the features



- CCDF of **Subscriber** ∩ **Subscriber** shows.
- Privacy? Protocol 2.
- Security? Hard to prevent fake accounts.

Privacy Analysis of the features



- CCDF of $\textbf{Subscriber}^s \cap \textbf{Subscription}^r$ shows less overlap among the subscriptions of authors of messages and subscriptions of the potential victims when the message is marked abusive.
- Privacy? Protocol 2.
- Security? Looks good!

Protocol 2: Private Set Intersection with Subscriber Signatures

- Suppose subscribers are willing to *sign* that they are subscribed.
- We still want the subscriptions to be private!
- BLS (Boneh et. al) signatures are compatible with our blinding.

$\Rightarrow$ Integrate them with our cut & choose version of the protocol.

Detailed protocol is in the paper.

What is Protocol 2 useful for?

- Prove overlap of subscribers without revealing their identity
- Key authentication in non-public Web-of-Trust (1-hop only)
- Unlike PSI of De Cristofaro (2016), no need for a CA!

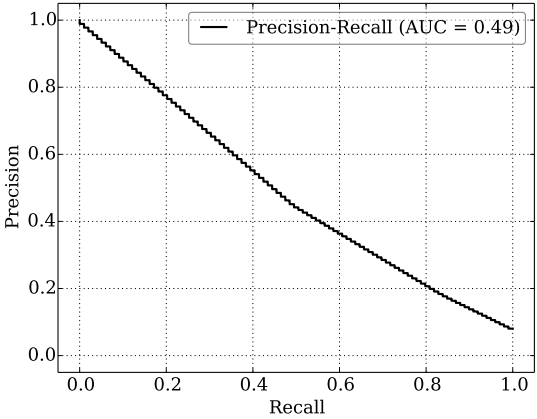
Detect Abuse

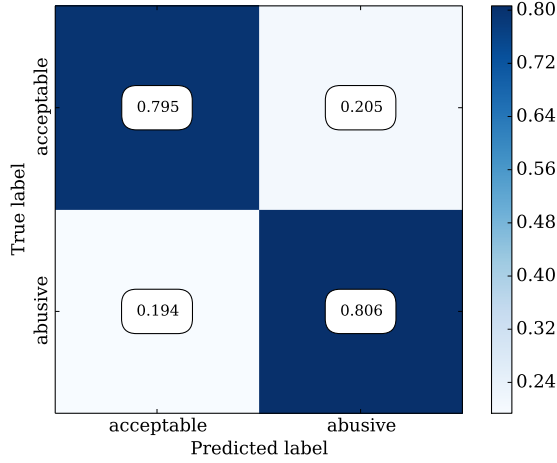
	Feature	Falsification/Adaptation	Crypto helps?
5.1	# lists	trivial	n/a
	# subscriptions	trivial	n/a
	$\frac{\#\text{ subscriptions}}{\text{age}}$	trivial	n/a
	$\frac{\#\text{subscriptions}}{\#\text{subscribers}}$	trivial	n/a
5.2	# mentions	costly	n/a
	# hashtags	costly	n/a
5.3	**message invasive**	**hard**	**n/a**
5.4	$\frac{\#\text{ messages}}{\text{age}}$	costly	yes
	# retweets	costly	n/a
	# favorited messages	costly	n/a
5.5	**age of account**	**hard**	**yes**
5.6	# subscribers	possible	minimally
	$\frac{\#\text{ subscribers}}{\text{age}}$	possible	minimally
5.7	subscription $\cap$ subscription	costly	w. privacy
5.8	subscriber $\cap$ subscriber	possible	w. privacy
5.9	**subscriber**^{s} $\cap$ **subscription**^{r}	**very hard**	**yes**
	subscription^s $\cap$ subscriber^r	possible	w. privacy

Little Data features shown in **bold**.

Extra Trees

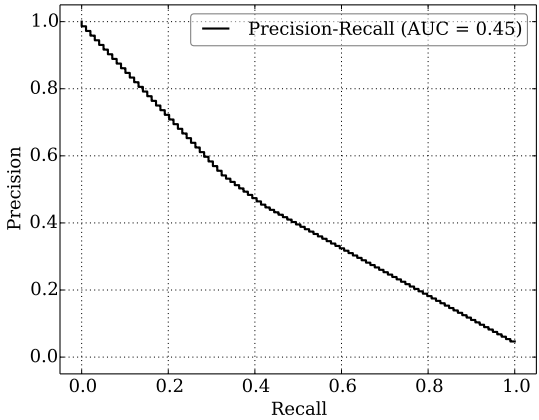
Only little data features

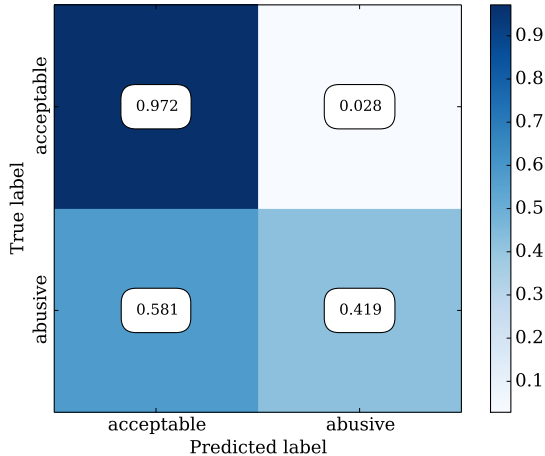




Gradient Boosting

Only little data features





Little Data Score

Classifier	Metric	Arithmetic Mean	Geometric Mean	Only Acceptable	Only Abusive
DT	Precision	0.64 ± 0.09	0.54 ± 0.04	0.98 ± 0.01	0.30 ± 0.17
	Recall	0.78 ± 0.12	0.76 ± 0.14	0.91 ± 0.08	0.64 ± 0.26
	F-score	0.67 ± 0.11	0.62 ± 0.09	0.95 ± 0.05	0.40 ± 0.18
RF	Precision	0.67 ± 0.12	0.59 ± 0.05	0.98 ± 0.01	0.36 ± 0.24
	Recall	0.76 ± 0.08	0.74 ± 0.09	0.94 ± 0.09	0.58 ± 0.19
	F-score	0.69 ± 0.12	0.64 ± 0.10	0.96 ± 0.05	$\mathbf{0.43} \pm 0.20$
ET	Precision	0.58 ± 0.05	0.40 ± 0.04	$\mathbf{0.99} \pm 0.02$	0.16 ± 0.08
	Recall	$\mathbf{0.80} \pm 0.17$	0.79 ± 0.16	0.79 ± 0.08	$\mathbf{0.80} \pm 0.33$
	F-score	0.58 ± 0.08	0.49 ± 0.08	0.88 ± 0.05	0.27 ± 0.13
GB	Precision	$\mathbf{0.71} \pm 0.10$	0.66 ± 0.04	0.97 ± 0.01	$\mathbf{0.45} \pm 0.20$
	Recall	0.70 ± 0.07	0.64 ± 0.07	$\mathbf{0.97} \pm 0.03$	0.42 ± 0.15
	F-score	$\mathbf{0.70} \pm 0.08$	0.64 ± 0.05	$\mathbf{0.97} \pm 0.02$	0.42 ± 0.14

Conclusions

- Method can protect privacy.
- Method can handle adaptive adversary.
- Little Data almost as good as humans!

Conclusions

- Method can protect privacy.
- Method can handle adaptive adversary.
- Little Data almost as good as humans!

But how to get this privacy onto the Internet?

Part III: No Such Data[4]

"When governments fear the people, there is liberty. When the people fear the government, there is tyranny. The strongest reason for the people to retain the right to keep and **bear arms** *is, as a last resort, to protect themselves against tyranny in government."*

—Thomas Jefferson

[4] Joint work with Jeffrey Burdges

Asynchronous messaging

Email with GnuPG provides authenticity and confidentiality...

- ... but fails to protect meta-data
- ... and also fails to provide *forward secrecy* aka *key erasure*

Why forward secrecy?

Imagine Eve records your GnuPG encrypted emails *now*, say here:

If Eve *ever* compromises your private key in the *future*, then she can read the encrypted emails you sent *today*.

Forward secrecy

Synchronous messaging

XMPP/OtR over Tor

- Forward secrecy from OtR
- User-friendly key exchange
- Location protection (Tor)
- ... but not asynchronous
- ... and leaks meta-data
- ... and not post-quantum

TOP SECRET//COMINT//REL TO USA, AUS//20320108

PWYA201207613540900007864O4

SIGAD: US-984XN
PDDG: AX
CASE_NOTATION: P2BSQC110024003
DTG: 16MR1345Z12

Active User [redacted]
Active User IP Address [redacted]
Target User [redacted]
Target User IP Address [redacted]
Start Mar 16, 2012 13:40:04 GMT
Stop Mar 16, 2012 13:44:46 GMT

Other User IP Addresses
[redacted]

Time (GMT)	From	To	Message
Mar 16, 2012 13:40:04	[redacted]		
Mar 16, 2012 13:40:28			
Mar 16, 2012 13:40:36			
Mar 16, 2012 13:40:43			
Mar 16, 2012 13:41:42			
Mar 16, 2012 13:41:58			[OC: No decrypt available for this OTR encrypted message.]
Mar 16, 2012 13:42:40			[OC: No decrypt available for this OTR encrypted message.]
Mar 16, 2012 13:43:42			[OC: No decrypt available for this OTR encrypted message.]
Mar 16, 2012 13:43:49			[OC: No decrypt available for this OTR encrypted message.]
Mar 16, 2012 13:43:55			[OC: No decrypt available for this OTR encrypted message.]
Mar 16, 2012 13:43:59			[OC: No decrypt available for this OTR encrypted message.]
Mar 16, 2012 13:44:20			[OC: No decrypt available for this OTR encrypted message.]
Mar 16, 2012 13:44:46			[OC: No decrypt available for this OTR encrypted message.]

***

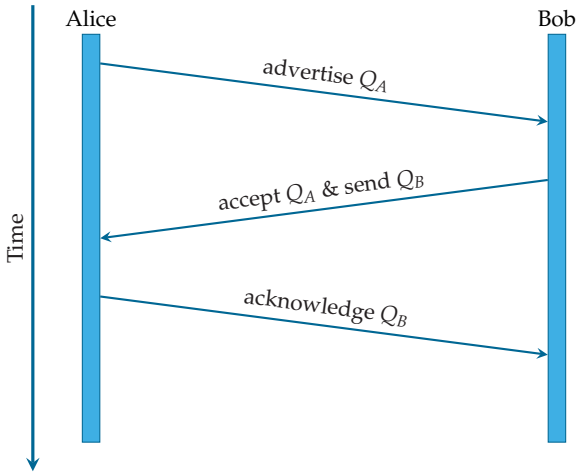
TOP SECRET//COMINT//REL TO USA, AUS//20320108

TOP SECRET//COMINT//REL TO USA, AUS//20320108

Why is OtR synchronous only?

We achieve *forward secrecy* through *key erasure* by negotiating an ephemeral session key using Diffie-Hellman (DH):

$$A^b = (g^a)^b = (g^b)^a = B^a \bmod p$$
$$d_A Q_B = d_A d_B G = d_B d_A G = d_B Q_A$$



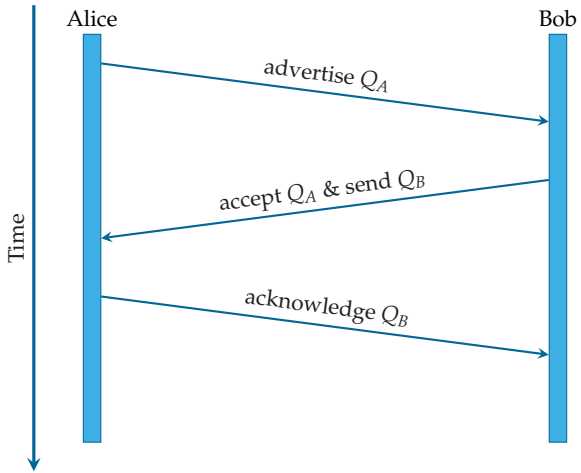
Private keys:
d_A, d_B

Public keys:
$Q_A = d_A G$
$Q_B = d_B G$

Why is OtR synchronous only?

We achieve *forward secrecy* through *key erasure* by negotiating an ephemeral session key using Diffie-Hellman (DH):

$$A^b = (g^a)^b = (g^b)^a = B^a \bmod p$$
$$d_A Q_B = d_A d_B G = d_B d_A G = d_B Q_A$$

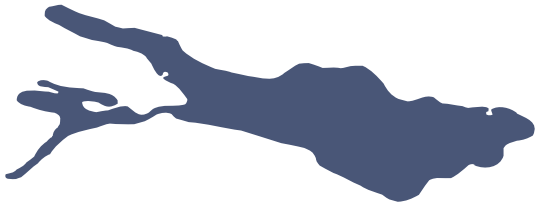


Private keys:
d_A, d_B

Public keys:
$Q_A = d_A G$
$Q_B = d_B G$

All three messages of the DH key exchange must complete before OtR can use a new ratchet key!

Introducing Project Lake[5]



[5] A lake is a big Pond.

Introducing Project Lake

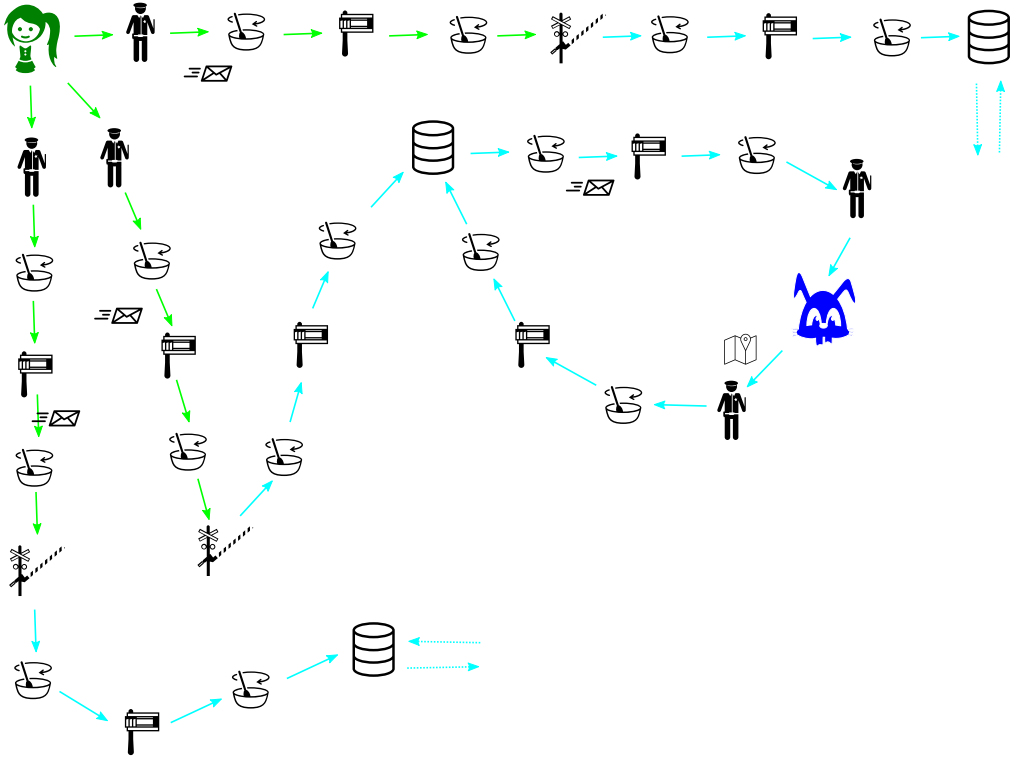
Layers:

MTA	IM
p≡p	
Lake	
Xolotl	
CADET	GNS
GNUnet-CORE	
TCP/IP	
Ethernet	

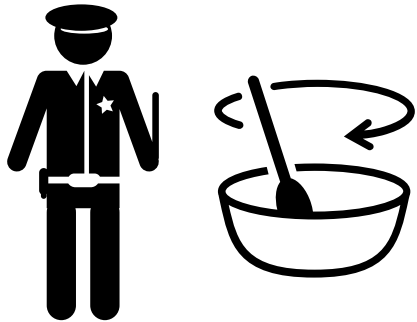
Properties:

- Endpoint **anonymity**
- Timing-attack resistance (mix, not circuit)
- No single point of failure: replicated mailbox
- Forward secrecy
- Post-quantum security
- Asynchronous delivery
- No meta-data leakage
- Off-the-record or on-the-record
- High latency

Lake Network Architecture



Asynchronous Mixing



Mixing vs. Onion Routing

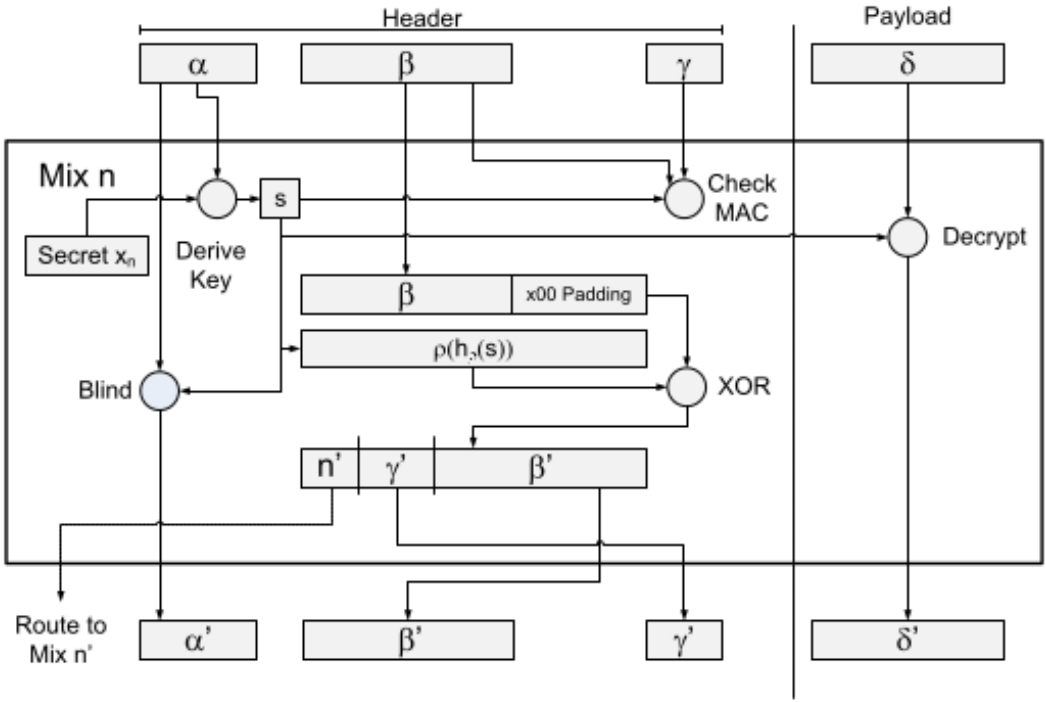
Onion routing:

- Source routing
- Circuit switching
- Low latency
- Vulnerable to timing attacks
- KX prevents replay attacks

Mixing:

- Source routing
- Packet switching
- High latency (message pool!)
- Timing attacks much harder
- Key rotation to prevent replay attacks

Sphinx by George Danezis and Ian Goldberg



The processing of a Sphinx message $((\alpha, \beta, \gamma), \delta)$ into $((\alpha', \beta', \gamma'), \delta')$

Sphinx properties

Provably secure in the universal composability model
[Camenisch & Lysyanskaya '05, Canetti '01]

1. Provides correct onion routing
2. Integrity, meaning immunity to long-path attacks
3. Security, including:
   - wrap-resistance[6]
   - indistinguishability of forward and reply messages

Replay protection implemented by Bloom filter (and key rotation).

[6] Prevents nodes from acting as decryption oracle.

Problem

Sphinx has forward secrecy only after key rotation.

- Long key lifetime:
  - Big Bloom filters to keep around to prevent replay attacks
  - Long window for key compromise
- Short key lifetime:
  - Limited delivery window after which messages are lost
  - Reduced mix effectiveness due to short time in pool
  - Loss of contact if reply addresses (SURBs) become invalid

Asynchronous Mixing with Forward Secrecy

Asynchronous Forward Secrecy with SCIMP

Idea of Silence Circle's SCIMP:

Replace key with its own hash.

Good:

New key in zero round trips.

Bad:

Once compromised, stays compromised.

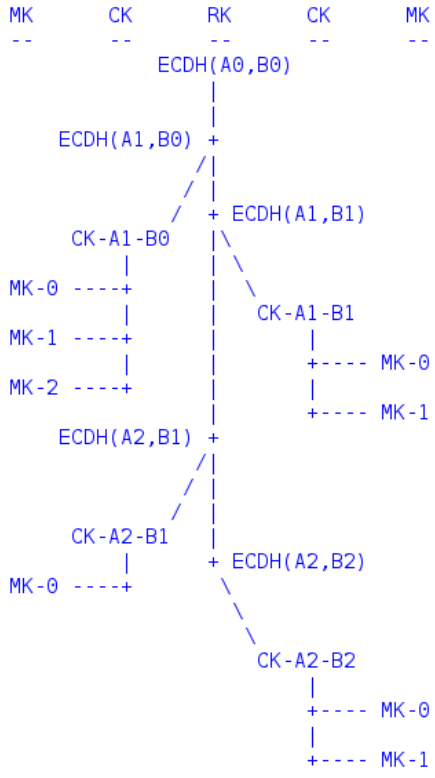
Axolotl by Trevor Perrin and Moxie Marlenspike

Approach:

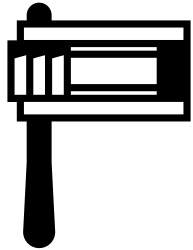
- Run DH whenever possible
- Iterate key by hashing otherwise
- Use TripleDH for authentication with deniability.

Result:

- Pseudonymous asynchronous KX
- Forward-secrecy
- Future secrecy
- Off-the-record
- Supports out-of-order messages
- Neutral against Shor's algorithm
- Formal security proof exists



Xolotl ≈ Sphinx + Axolotl



Ratchet for Sphinx

Can we integrate a ratchet with Sphinx?

Axolotl does not work directly because:

- Relays never message users
- Cannot reuse curve elements

Idea:

- Users learn what messages made it eventually
- This is particularly true for replies

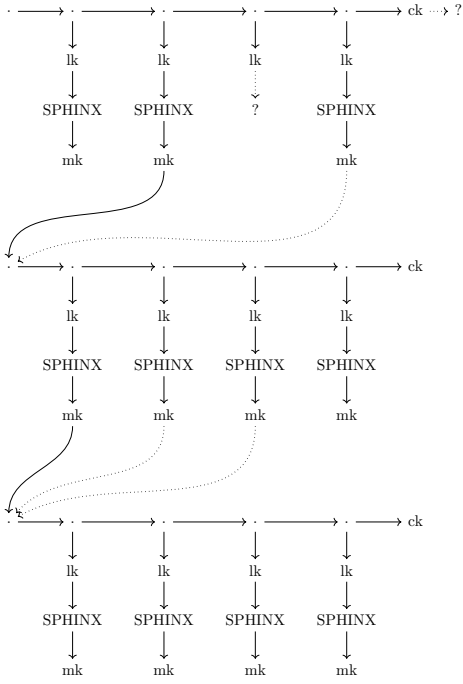
Client directs mix's ratchet state

Acknowledging ratchet state

Chain keys evolve like Axolotl, producing leaf keys.

Create message keys by hashing a leaf key with a Sphinx ECDH

$$\mathrm{mk} = H(\mathrm{lk}, H'(\mathrm{ECDH}(\mathrm{u}, \mathrm{r})))$$



Acknowledging ratchet state

Chain keys evolve like Axolotl, producing leaf keys.

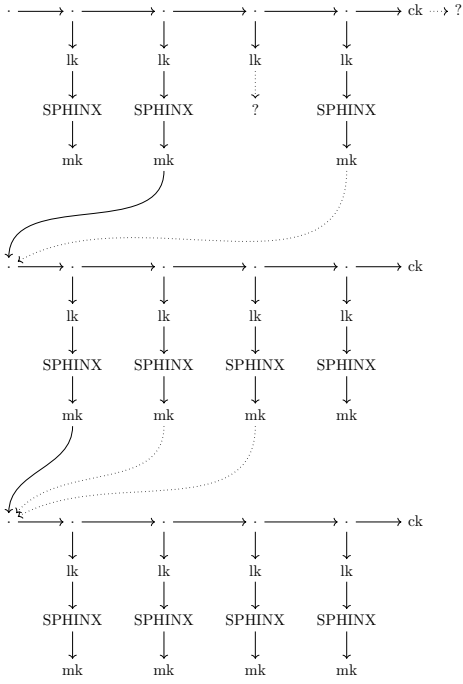
Create message keys by hashing a leaf key with a Sphinx ECDH

$$\mathrm{mk} = H(\mathrm{lk}, H'(\mathrm{ECDH}(\mathrm{u}, \mathrm{r})))$$

Packets identify the message key from which their chain started.

And their leaf key sequence no.

And parent max sequence no.



Ratchet placement

We cannot use the Xolotl ratchet for every mixnet hop:

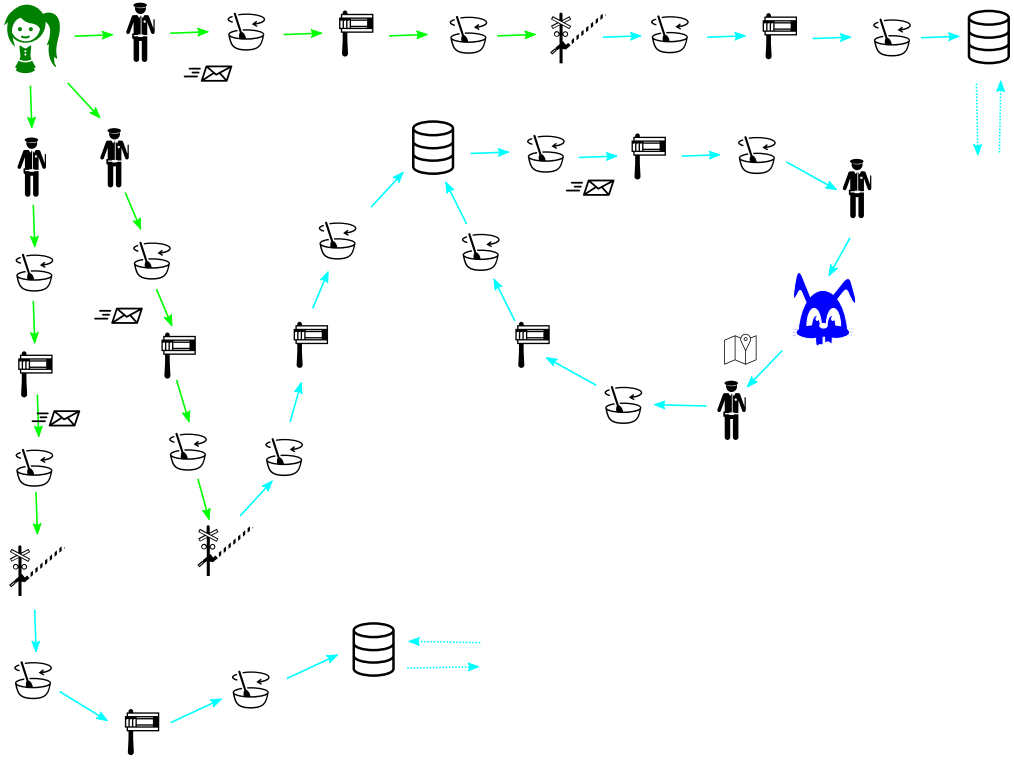
- Use of ratchet state results in pseudonymity
- Setup of post-quantum KX may be excessively expensive

Safe places:

- Third hop out of a five hope circut (long-term ratchet)
- Guard node (while connection is maintained)

Other hops should use “ordinary” mix.

Lake Network Architecture



Hope

"However, minority groups can influence the majority by showing a sense of **consistency***; demonstrated* **investment***;* **independence***;* **balanced judgment***; and similarity to the majority in terms of age, gender and social category."*

—TOP SECRET JTRIG Report on Behavioural Science

Further reading

1. Christian Grothoff, Bart Polot and Carlo von Loesch. *The Internet is broken: Idealistic Ideas for Building a GNU Network*. **W3C/IAB Workshop on Strengthening the Internet Against Pervasive Monitoring (STRINT)**, 2014.
2. Álvaro García-Recuero, Jeffrey Burdges and Christian Grothoff. *Privacy-Preserving Abuse Detection in Future Decentralised Online Social Networks.* **Data Privacy Management (DPM)**, pages 78–93, 2016.
3. Jeffrey Burdges and Christian Grothoff. *Xolotl-Lake*. Available in the future and in `lake.git`. 2018?
4. Neal Walfield and Christian Grothoff. *TomorrowToday: GSM-based Location Prediction*. Available upon request. 2016.
5. Phillip Rogaway. *The Moral Character of Cryptographic Work*. **Asiacrypt**, 2015.