Anonymous Traitor Tracing

Mark Zhandry

Join work with Ryo Nishimaki, Daniel Wichs
Traitor Tracing

Goal: Using leaked key, identify traitor to revoke key, punish, disincentivize
Considerations

- What’s wrong with $u = (\mathcal{K}, u)$?
- What if adversary obfuscates $\text{Dec}(u, \cdot)$?
- What if broken key that only recovers half the message?
  - Assume traitor produces pirate decoder: \(\mathcal{P}: c \rightarrow \{0,1\}\)
  - Only given oracle access to \(\mathcal{P}\)

- What if 2 spys? \(k\) spys?
  - Allow adversary to get arbitrarily many secret keys
    (Bounded collusion also interesting)
Syntax

**Setup()**: Outputs \((msk, pk)\)

**Enc(pk, m)**: Outputs a ciphertext \(c\)

**KeyGen(msk, u \(\in [N]\))**: Outputs user \(u\)’s secret key \(u\)

**Dec(u \(\rightarrow\), c)**: Outputs \(m\)

**Trace\(\rightarrow\)(pk)**: Outputs an “accused set” \(A \subseteq [N]\)
Properties

Correctness: $\text{Dec}(u, \text{lock}) = \text{lock}$ for all $u$

Semantic Security: w/o any $u$, $\text{lock}$ hides $\text{folder}$

Traceability:
- $A \setminus T = \emptyset$
- If $\text{skull} \text{ "usefull" (breaks } \text{lock}, \text{ then } A \neq \emptyset$
A Trivial System

Each user gets own public key/secret key for PKE scheme

Ciphertext = encryption under each public key

Tracing: encrypt $m$ under several public key, junk for others
  * Successful decryption $\rightarrow$ Traitor

Limitation: parameter sizes, running times grow with $N$

Goal: minimize $|c|$, $|pk|$, $|u\rightarrow|$, $|msk|$
  (Also, handle exponential $N$)
Prior Work

**Combinatorial** (CFN’94, …)
- Bounded collusion $k$
- Very weak generic assumptions (OWF, PKE)
- State of the art: $|c|, |pk|, |u\rightarrow| = \text{poly}(k, \log N)$

**Algebraic** (BF’99, BSW’06, …)
- Bounded or unbounded collusion
- Specific assumptions (DDH, Subgroup Decision)
- State of the art for unbounded: $|c|, |pk|, |u\rightarrow| = O(N^{\frac{1}{2}})$

**Obfuscation-Based** (GGHRSW’13, BZ’14)
- Generally always unbounded collision
- Extremely strong assumptions (iO, FE)
- State of the art: $|c|, |pk|, |u\rightarrow| = \text{polylog}(N)$
Who Keeps Track of User Info?

After tracing, get index $u$ of user (integer from 1 to $N$)

- Sufficient for revocation
- How to prosecute? Maintain database:

  \[
  \begin{align*}
  u=1 & \rightarrow \text{Address 1, Credit card number 1} \\
  u=2 & \rightarrow \text{Address 2, Credit card number 2} \\
  \vdots
  \end{align*}
  \]

This approach: ability to punish implies lack of anonymity

Q: Are tracing an anonymity at odds?
Embedding Arbitrary Info in Key

Why not set \( u = \) “Address, Credit card number”?

- Length of identifying info \( L \rightarrow N = 2^L \)

- Current systems: \( N \) polynomial
  \( \rightarrow L \) is logarithmic

- To embed arbitrary info, need exponential number of identities
Anonymity

\[ u = \text{Ident. info} \]

\[ \text{MPC} \]

\[ \text{msk} \]

\[ \emptyset \]

Verification authority
Previous Traitor Tracing

Formula for essentially all schemes with unbounded collusions:

Private Linear Broadcast Encryption (PLBE) + Generic Tracing Algorithm [BSW’06] = Traitor Tracing (w/ same params)
Private Linear Broadcast Encryption

**Functionality:** encrypt to intervals

**Security:** as little info about interval leaked as possible
Private Linear Broadcast Encryption

ID = {1, ..., N}

Setup(): Outputs (msk, pk)

Enc(pk, m, v ∈ [0,N]): Outputs a ciphertext c

KeyGen(msk, u ∈ [N]): Outputs user u’s secret key

Dec(u, c): Outputs m
Properties of PLBE

Correctness: \( \text{Dec}(u, (\text{\boldsymbol\Delta} , v)) = \text{\boldsymbol\Delta} \) if \( u \leq v \)

Semantic Security:
\( \text{Enc}(pk, (\text{\boldsymbol\Delta} , 0)) \) reveals no info about \( \text{\boldsymbol\Delta} \)
even given many \( u \)

Recipient privacy:
Cannot distinguish \( \text{Enc}(\text{\boldsymbol\Delta} , u) \) from \( \text{Enc}(\text{\boldsymbol\Delta} , u-1) \)
unless you know \( u \)
TT from PLBE

\[ f(v) = \Pr[ \text{decrypts ( , v)} ] \]

PLBE security \( \rightarrow \) \( \delta \) negligible

Decoder functionality \( \rightarrow \) \( \epsilon \) "large"
Tracing PLBE [BSW’06]

\[ p_0 = \Pr[\text{folder unlocked}] \]

\[ p_1 = \Pr[\text{folder locked}] \]

\[ \ldots \]

\[ p_N = \Pr[\text{folder unlocked}] \]

Output any \( u \) for which \(|p_{u-1} - p_u|\) is large.
Large-Identity Traitor Tracing from PLBE

Private Linear Broadcast Encryption (PLBE)

[GGHRSW'13, BZ'14]
From iO/FE

Generic Tracing Algorithm [BSW'06]

Tracing time $\text{poly}(N)$

=\n
Traitor Tracing (w/ same params)
Given oracle access to $f: [0,N] \rightarrow [0,1]$

- Several “jumps”
- Between jumps, $f$ varies minimally
- At jump, arbitrary change
- $f(0)$ small, $f(N)$ large (implies noticeable change at some jump)

Goal: Find location of one of the jumps
Oracle Jump Finding

BSW’06 alg → Linear search to find jump

• Visits every point, so running time $O(N)$

For efficient tracing of large $N$, need running time $\text{polylog}(N)$

• Can’t visit every point in domain
Binary Search?

\[ \Delta < \delta \]  \quad  \Delta > \epsilon - 2\delta

\[ \Delta < \delta \]  \quad  \Delta > \epsilon - \delta \]
Binary Search?

Which side do I recurse on?
• Larger gap?  
  Gap decreases by ½ each time  
  Gap doesn’t tell us how many jumps
• Both?  
  Still polynomial time in $\log(N)$?
Always recurse on gap

Alg from [BCP’14], entirely different context

Question: why guaranteed to be polynomial time?
Tree of Intervals

[0, N]

[0, N/2]  (N/2, N]

Recurse on interval  →  jump in that interval
Large-Identity Traitor Tracing from PLBE

Private Linear Broadcast Encryption (PLBE)

[GGHRSW’13, BZ’14]
From iO/FE

New Generic Tracing Algorithm [BCP’14, NWZ’15]

Tracing time $\text{polylog}(N)$

Traitor Tracing (w/ same params)
Limitations of PLBE Approach

Suppose I want to embed much more info into key
• User ID = Name + Address + Map + Picture/Video + …

Given $\text{msk}$, can recover $v$ from $(\text{lock}, v)$
• Find $v'$ s.t. $v'$ decrypts $\text{ctxt}$, $v'$ but does not

Given $\text{pk}$, can recover $u$ from $u$
• Find $u'$ s.t. $u'$ decrypts $(\text{lock}, u')$, but not $(\text{lock}, u'-1)$

PLBE: $|\text{ctxt}|$, $|u-1| \geq \log N = |$ identifying info $|

Q: Is this inherent to Traitor Tracing?
Limitations of Traitor Tracing

Given $pk, u\rightarrow$, recover $u$: trace $\mathbb{P}$ = Dec($u\rightarrow, \cdot$)

$TT: |u\rightarrow| \geq |\text{identifying info}|$

For ctxt size, apparently no such restriction

To get small ciphertexts, need alternative to PLBE
Private Block Linear Broadcast

Identifying info encoded as curve
**Private Block Linear Broadcast**

**Functionality:** can decrypt if point “to the right” of curve

**Security:**
- Can’t decrypt if point “to the left” of curve
- Can’t learn anything about ⬤ except “left” or “right”

**Ctxt:** (⚠️, ⬤)

Diagram:
- Arrows pointing to the right indicate decryption is possible.
- Arrows pointing to the left indicate decryption is not possible.
- The green circle represents the point being decrypted.
- The red circle represents a point not being decrypted.

Axis labels:
- r (left)
- 0
- N (right)
**Theorem:** Can trace as long as
- Curves do not intersect
- Curves confined to oscillate about a single column

Size of info encoded by curve: \( \geq r \)

Info encoded in ctxt:
- \(| \text{message} | + \log r + \log N \)

Ctxts only need to grow logarithmically with embedded info
- Can achieve from obfuscation using [AS’15]
Small variation $\delta$ between curves
Large variation $\epsilon$ across domain
→ Large jump at some curve
→ Gives rise to generalization of Jump Finding Problem
Conclusion

First traitor tracing system to handle exponential number of user identities

- Allows for “identity based” traitor tracing
- Allows for anonymity + tracing to coexist
- Can embed arbitrarily large info into key w/o affecting ctxt size
- Also show how to revoke

Main open question:

TT from weaker assumptions (MMaps, lattices, etc)

Thanks!