UCBlocker: Unwanted Call Blocking Using Anonymous Authentication

Changlai Du, Hexuan Yu, Yang Xiao,
Y. Thomas Hou, Angelos D. Keromytis, Wenjing Lou
Spam and Scam Calls in the US

Monthly Spam Received

<table>
<thead>
<tr>
<th>Year</th>
<th>Spam Calls</th>
<th>Spam Texts</th>
</tr>
</thead>
<tbody>
<tr>
<td>2015</td>
<td>10.9</td>
<td>6.3</td>
</tr>
<tr>
<td>2017</td>
<td>18.8</td>
<td>8.4</td>
</tr>
<tr>
<td>2018</td>
<td>25</td>
<td>8.5</td>
</tr>
<tr>
<td>2019</td>
<td>31.7</td>
<td>10.6</td>
</tr>
<tr>
<td>2020</td>
<td>28</td>
<td>14.7</td>
</tr>
<tr>
<td>2021</td>
<td>30.7</td>
<td>16.9</td>
</tr>
<tr>
<td>2022</td>
<td>31</td>
<td>16.5</td>
</tr>
</tbody>
</table>

19.5/ month

Source: Truecaller Insights/Harris Poll

Total Money Lost to Scam Calls

<table>
<thead>
<tr>
<th>Year</th>
<th>$M</th>
</tr>
</thead>
<tbody>
<tr>
<td>2014</td>
<td>8.6</td>
</tr>
<tr>
<td>2015</td>
<td>7.4</td>
</tr>
<tr>
<td>2017</td>
<td>9.5</td>
</tr>
<tr>
<td>2018</td>
<td>8.9</td>
</tr>
<tr>
<td>2019</td>
<td>10.5</td>
</tr>
<tr>
<td>2020</td>
<td>19.7</td>
</tr>
<tr>
<td>2021</td>
<td>29.8</td>
</tr>
<tr>
<td>2022</td>
<td>39.5</td>
</tr>
</tbody>
</table>

$39.5 B

Source: Truecaller Insights/Harris Poll

Enablers of the Spam/Scam Call Problem

- **VoIP (Voice over Internet Protocol) + Autodialers**
  - Massive calls at very low cost
  - Over the Internet, cross jurisdictions

- **Caller ID Spoofing**
  - Altering the Caller ID field (phone number and/or name) is easy
  - Spoofing legit government agencies/businesses
Existing Spam/Scam Call Defenses

- **End-to-end Authentication**
  - Via voice channel: **Authloop** [Security’16]
    - ~9 seconds due to low bandwidth (300 to 3400 Hz)
  - Via data channel: **AuthentiCall** [Security’17]
    - 1-1.4 seconds
    - Require a trusted server

- **Network-assisted Solution** - **STIR/SHAKEN** [FCC’20]
  - Caller ID authentication and verification over IP networks

Only prevent caller ID spoofing, but still not all the unwanted calls that utilize legitimate caller IDs.
Our Solution - UCBlocker (1/2)

- **User-defined Policy**
  1) Callee can set up attribute-based caller authentication policies
  2) Enables incoming calls from legitimate unknown numbers

- Utilize **Attribute-based Anonymous Credentials (AC)**

```json
{"policies": [{"ZIP = 94550": TRUE, "schema": "did:schema:abcde"}]}"
```
Our Solution - UCBlocker (2/2)

- Decouples end-to-end caller authentication from call session initiation
  - Authentication - Out-of-Band
  - Call Session initiation over telephone networks

- One-time Verification Code
  - Binding authentication and call session
  - Sent for call-time verification
Anonymous Credentials (AC)

allows users to prove that they satisfies certain properties *without* disclosing unnecessary information

Cryptographic Primitives:

- Zero-Knowledge Proof (ZKP)
- ZKP-friendly signature schemes (e.g., BBS+)
- Commitment Schemes (e.g., Pedersen)
AC and Anonymous Presentation

- One AC can contain a set of attributes
- One **caller** can hold multiple ACs that issued by different **issuers**

Original Credential

Anonymous Presentation

1 AC → n verifiable presentations (Indistinguishable)

- Selective Disclosure - prove knowledge of hidden attributes
- Prove the integrity, authenticity of the AC

✅ Caller Privacy
Who can issue the credentials?

Issuers can be different entities, e.g.,

1) **Callee** - Issue **Contact Credentials** to their friends through Internet (e.g, Facebook Messenger)

2) **Trusted Authority** - e.g., a Digital Driver License issued by DMV

3) **MVNO** (Mobile Virtual Network Operator, e.g., Google Fi) - a dedicated UCBlocker service provider/carrier
Example - Legitimate Unknown Caller (1/2)

0. Credential Issuance

Original Credential

Digital Wallet

Alice (Caller)

1. Lookup Bob’s policy entry according to his phNum

Bob’s Policy

{"policies": [{ "ZIP = 94550": TRUE, "schema": "did: schema: abcde" }]}

Bob (Callee)

Anonymous Presentation

Only Reveal the attribute “ZIP”
Verifiable and Unlinkable

Verifier

2. Show a one-time presentation

Data registry
Example UI Interfaces - Policy Define and Attribute disclosure

Bob's Device

Define a policy with the attributes and predicates

<table>
<thead>
<tr>
<th>Zip Code</th>
<th>Predicate</th>
<th>20001</th>
</tr>
</thead>
<tbody>
<tr>
<td>equal</td>
<td>conjunct</td>
<td></td>
</tr>
<tr>
<td>unequal</td>
<td>disjunct</td>
<td></td>
</tr>
<tr>
<td>in_set</td>
<td>in_range</td>
<td></td>
</tr>
</tbody>
</table>

Set

Alice's Device

Do you want to share the following attributes with Bob for authentication?
zip code

Agree  Decline

Alice's Device

Please select an attribute from a credential schema

- Issuer_MVNO
- Issuer_DMV
  - Schema_State_ID
  - Schema_Driver's_License
    - Attribute_Identifier
    - Attribute_Name
    - Attribute_Address
    - Attribute_City
    - Attribute_State
    - Attribute_Zip_Code
    - Attribute_Date_of_Birth
Example - Legitimate Unknown Caller (2/2)

0. Credential Issuance

1. Lookup Bob’s policy entry according to his phNum

2. Show a one-time presentation

3a. VeriCode
3b. VeriCode

4. Call initiation with VeriCode

5. Accept \( \in \{\text{VeriCode}\} \)

Bob (Callee)

Issuer

Verifiable data registry

VeriCode

VeriCode

The Out-of-band authentication is done!
3 Methods of Transmitting Verification Code

1) Add an extra header field in SIP signaling message
   - Similar to STIR/SHAKEN
   - Requires substantial investment from all stakeholders

2) Using Voice Channel
   - ~300 ms for a 128-bit verification code transmission (500 bps channel)

3) Repurposing Caller ID (of SIP)
   - Replace the caller ID with our VeriCode
   - Can be easily set by a VoIP client or connected PBX in the header field
   - 32-bit VeriCode - no extra cost
Evaluation - Implementation

- **VoIP PBX running on an AWS instance**
  - PBX connects to the telephone networks using SIP trunk services

- **UCBlocker Client**
  - Issuer, User, Verifier
  - Anonymous credentials
    - Relic toolkit
    - libpabc
    - BLS12-381 Elliptic Curve
    - Libsodium

- **Verifiable data registry**
  - Public ledger - Hyperledger Indy
Evaluation - Time Consumption

- ~1.5s end-to-end delay for a successful authentication
  - Lookup -> Proof construction -> Proof verification
  - VeriCode issuance

(a) Credential issuance  
(b) Proof generation  
(c) Verification
Summary

- **Flexibility**
  - Only calls that follow the callee’s policies can reach to the callee
- **Usability**
  - Legitimate calls from unknown numbers is supported
- **Privacy**
  - Caller does not need to disclose unnecessary information for authentication
- **Compatibility**
  - Minimal changes to the telephone networks
  - Eliminates the need for a call-time data channel
- **Efficiency**
  - No significant delays to original call session setup
Thank you for your attention!

Q&A