Blind Signatures: Past, Present, and Future

Julian Loss
ECash Systems (Chaum 1983 [C83])

- User wants to trade physical for digital currency
- Challenge: retain the **physical** attributes of cash
  - Unforgeable (no double spending)
  - Untraceable (pecunia non olet)
Solution: Blind Signatures

- User: create identifier \( ID \)
- Bank: sign \( ID \) with signature \( \sigma \)
- User: derive coin from \( \sigma \)
- **Blindness**: Bank does not see \( ID \)
  - \( \implies \) Coin looks “unrecognizable” to bank
More Applications of Blind Signatures

- EVoting: Voting authority blindly signs ballot for approval
- Anonymous Credentials: Credential authority blindly signs credential for validation
- More recently: Blockchain Applications, e.g. Coin Shuffling/Mixing
Blind Signatures: Syntax

- Signer holds secret key $sk$
- User holds public key $pk$, message $m$
- Signer generates signature $\sigma$ on $m$
Blind Signatures: Security Properties

- **Blindness**: Signer does not learn $m$
  - Should hold even if Signer generates keys
- **Unforgeability**: User cannot create $\sigma$ by itself
- How to formalize these properties?
Signer cannot determine $b$

$\implies$ Signatures cannot be linked to signing sessions
One-More Unforgeability

- Signer and User engage in $\ell$ sessions
  - $\rightarrow$ User obtains $\ell$ signatures
- User cannot generate $\ell + 1$ signatures
- Very strong security property
  - $\rightarrow$ Difficult to construct
The History of Blind Signatures

- This talk: overview of the state of the field; what question are open?
- Complicated history: Bugs, attacks, forgotten papers, and more
- Active field of research with major improvements still being made
Simple Analogy: Speedrunning

- **Speedrunning**: beating a video game as fast as possible
- What constitutes a valid run? Rules?
- For blind signatures, we care about signature sizes and communication, model assumptions

<table>
<thead>
<tr>
<th>Glitchless</th>
<th>No Major Glitches (a.k.a. Memory Corruption)</th>
<th>Any %</th>
</tr>
</thead>
<tbody>
<tr>
<td>1:45:05</td>
<td>0:11:33</td>
<td>0:01:18.893</td>
</tr>
</tbody>
</table>

- Plain model
- Random Oracle Model, Conservative assumptions
- Generic group model, strong assumptions
Cryptography relies heavily on number theory for problem with average-case hardness

Example: Factoring large numbers into prime components

Conjectured hardness \(\implies\) security of scheme

Ideally want conservative conjectures:

- Simplest version of problem (DLOG < CDH < DDH < ..., FAC < RSA < ...)
- Non-interactive, non-parametrized problems (non-examples: One-More DLOG/RSA, LRSW,...)
- Should have stood test of time (DLOG, CDH, DDH, FAC, RSA, QR, SIS, LWE,...)
Past I: 1983-2002
Based on the RSA Full-Domain Hash Signature Scheme

\[ N = P \cdot Q \text{ for primes } P \text{ and } Q, \]

\[ e \text{ is an integer s.t. } \gcd((P - 1) \cdot (Q - 1), e) = 1 \]

\[ pk = (N, e) \text{ and } sk = d \text{ s.t. } d \cdot e \equiv 1 \pmod{(P - 1) \cdot (Q - 1)} \]

Problem: unforgeability relies on very strong hardness assumption

\[ \sigma = u \cdot r^{-1} = H(m)^d \pmod{N} \]
1990: Schnorr’s Blind Signature Scheme

- Based on the Schnorr Signature Scheme
- Uses group $\mathbb{G}$ of prime order $p$ with generator $g$
  - $x \in \mathbb{Z}_p, X = g^x$
  - $sk = x, pk = X$
- Inherently “linear”

$R = g^{r+r'}$
$c = c' + \beta$
$s = c \cdot x + r + r'$

$\alpha, \beta \leftarrow \mathbb{Z}_p$
$c' = H(R \cdot g^\alpha \cdot X^\beta)$
$s' = s + \alpha, \sigma = (c', s')$
1996: Toward Provable Security

- Breakthrough Papers of Pointcheval and Stern [PS96,PS977,PS00]
- First formal definition of One-More-Unforgeability
- First provably secure blind signatures,
- Introduces rewinding in the Random Oracle Model
- Double generator variant of Schnorr’s Scheme (Okamoto-Schnorr)
- Caveat: only logarithmically many signatures per public key
Limitations of PS: A Closer Look

- $\ell$ = number of signatures
- $Q$ = number of hash queries
- $p$ is the group order

- Then, PS is secure as long as $Q^{\ell+1}/p \ll 1$
- For 256 bit prime $p$ and $Q = 2^{128}$, $\ell \geq 1$ makes PS theorem meaningless
- Explicit Question in PS: Is this limitation inherent?
Communication and Signature Sizes

- Assuming $2^{30}$ signatures, $2^{128}$ hash queries
- 2000:
  - Roughly $2^{37}$ bits (17.18GB)
  - RSA, FAC, DLOG
Schnorr’s ROS Problem [S01]

\[ A := \begin{bmatrix}
A_{1,1} & \ldots & A_{1,\ell}, \\
\vdots & \ddots & \vdots \\
A_{\ell+1,1} & \ldots & A_{\ell+1,\ell}, \\
\end{bmatrix}, \quad \begin{bmatrix}
H(A_1) \\
\vdots \\
H(A_{\ell+1})
\end{bmatrix}, \quad \begin{bmatrix}
c_1 \\
\vdots \\
c_{\ell} \\
-1
\end{bmatrix} \]

- \( \ell \)-ROS: Find \( A \in \mathbb{F}_p^{(\ell+1) \times (\ell+1)} \) and \( \vec{c} \in \mathbb{F}_p^{\ell+1} \) s.t. \( A \cdot \vec{c} = 0 \)

- Conditions on last column of \( A \) and vector \( \vec{c} \):
  - Last column of \( A \) is generated via random oracle \( H \)
  - Depends on the first \( \ell \) columns of \( A \)
  - Last column is a linear combination of \( \ell \) first columns of \( A \)

- Solution exists iff \( Q^{\ell+1}/p \geq 1 \)
Schnorr’s Attack

- Attacker opens $\ell = \log(p)$ concurrent sessions, gets $R_1, \ldots, R_\ell$
- Solves $\ell$-ROS problem relative to $\widetilde{H}$, where $\widetilde{H}(\vec{a}, u) = H(\Pi_{i=1}^\ell R_i^a_i, u)$
- Constructs $s'_i$ from $s_1, \ldots, s_\ell$ via ROS solution as $s'_i = \sum_{j=1}^\ell A_{i,j} \cdot s_j$, $c'_i = H(\Pi_{j=1}^\ell R_j^{A_{i,j}}, m_i)$
- Applies to Schnorr and OS Blind Signature Schemes
- Schnorr does not give an algorithm for ROS
The k-List Problem (Wagner 2002)

- Given $k$ random lists $L_1, \ldots, L_k$ find $x_1 \in L_1, \ldots, x_k \in L_k$ s.t. $\sum_i x_i \pmod{p} = 0$
- Generalization of Birthday Problem
- Wagner proposes **heuristic** algorithm with subexponential runtime [W02]
- Can be used to solve ROS in subexponential time, given there exists a solution
- Shows that PS-analysis is optimal!
Past II: 2003-2018
Interest in Blind Signatures Fades

- ECash and EVoting mostly thought of as theoretical concepts
- Some papers on anonymous credentials and blind signatures in plain model
  - Mostly of theoretical interest
  - Only sequential security (hard to use in practice)
  - Use “unreasonable” hardness assumption
- Exception: Rückert gives first (efficient) Lattice-based construction
  - Later shown to be flawed
- Result:
  - 15-year period of stagnation (relative to the explosion of the field)
  - Much of the literature is forgotten, people are not aware of ROS by 2018
Present: 2019-2022
2019: Blockchains?

- Blockchains lead to renewed interest in blind signatures
- At this point, almost no one:
  - Remembers the ROS attack
  - Understands the proofs of PS
- Two important papers 2019:
  - Modular formulations of PS papers [HKL19]
  - Applying ROS attack to various multisignature schemes [DEFKLNS19]
- Rekindles interest in blind signatures and raises awareness of ROS attack
2020: Lattices, mROS

- New Version of ROS: mROS [FPS20]
  - Potentially harder, easy to work with
  - No lower bounds (currently)
- Revisiting and correcting Rückert’s Lattice-based construction [HKLN20]
  - Still logarithmically bounded number of signatures
  - Still no progress toward efficient schemes from simple assumptions
2021: ROS, Boosting

- EUROCRYPT 2021: new algorithm for ROS [BLLOR21]
  - Polynomial time when $\ell \geq \log_2(p)$
  - Major improvements even for smaller $\ell$
  - Makes ROS attack actually practical

- ASIACRYPT 2021: revisits “boosting” construction of Pointcheval [KLR21]
  - Transforms any of the EUROCRYPT 2019 schemes into polynomially secure ones
  - Large communication overhead (linear in number of signatures)
  - Does not work for lattice constructions
Communication and Signature Sizes

- Assuming $2^{30}$ signatures, $2^{128}$ hash queries

2000:
- Roughly $2^{37}$ bits (17.18GB)
- RSA, FAC, DLOG

2021:
- Roughly 12000 bits (1.5KB)
- RSA, FAC, DLOG
2022: Efficient Boosting, More Lattice Constructions

- **Boosting:**
  - Exponentially improvements in communication overhead for boosting construction
  - Practical schemes from pairings
  - Computation still linear in number of issued signatures
  - Manuscript: reduces computation [HLW22]

- **Lattices:**
  - First practical lattice-based blind signature schemes [PK22]
Communication and Signature Sizes

- Assuming $2^{30}$ signatures, $2^{128}$ hash queries

- 2000:
  - Roughly $2^{37}$ bits (17.18GB)
  - RSA, FAC, DLOG

- 2021:
  - Roughly 12000 bits (1.5KB)
  - RSA, FAC, DLOG

- 2022
  - 3KB size, 120KB communication (CDH + pairings)
  - 9KB size, 8KB communication (RSA)
  - 100KB size, 850KB communication (NTRU)
Future: 2023-
Important Open Questions

- **Constructions:**
  - Pairing-free constructions
  - Blind signatures with strong compatibility (e.g., Bitcoin, Ethereum, etc.)
  - Lattices: communication still around 1MB per signature

- **Cryptanalysis:**
  - Polynomial-time ROS attack for lower dimensions?
  - Extend ROS attack to lattices
  - Prove (or disprove) attack on mROS [FPS20]
THANK YOU!
References

- PS97: David Pointcheval, Jacques Stern. New Blind Signatures Equivalent to Factorization. CCS 1997
- S01: Claus-Peter Schnorr. Security of Blind Discrete Log Signatures against Interactive Attacks. ICICS 2001
- W02: David A. Wagner. A Generalized Birthday Problem. CRYPTO 2002
References

- HKLN20: Eduard Hauck, Eike Kiltz, Julian Loss, Ngoc Khanh Nguyen. Lattice-Based Blind Signatures, Revisited. CRYPTO 2020