Partially Blind RSA Signatures

draft-amjad-cfrg-partially-blind-rsa

Ghous Amjad*, Scott Hendrickson, Christopher Wood, Kevin Yeo
IETF 120 - CFRG
Outline

• Motivation
  • Background: Blind RSA Signatures
  • Partially Blind RSA Signatures
  • Benchmarks
  • Current Status
Motivation: Blind Signatures

- Privacy Pass
- Web Browsing, e.g.,
  - Google VPN
  - iCloud Private Relay
- Avoiding Repeated CAPTCHA Solving
- Private Click Measurement
- Tor DOS Defenses
- …
Motivation: Partially Blind Signatures

• ‘draft-ietf-privacypass-public-metadata-issuance’
  • Adopted in the PrivacyPass working group
  • Defines two variants for token schemes with public metadata
    • Privately verifiable (‘draft-irtf-cfrg-voprf’)
    • Publicly verifiable (this draft!)

• Blind signatures enabling public metadata such as
  • expiration times,
  • service multiplexing etc.

• Avoiding one key per metadata approach
  • May require fixed public metadata choices ahead of time
  • Key management scalability concerns
Motivation: Blind RSA Signatures

• RFC 9474 for Blind RSA Signatures
  • Simple (one-round scheme, stateless server issuance)
  • Widely supported public verification

• Natural to standardize variant supporting public metadata
Outline

• Motivation
• Background: Blind RSA Signatures
  • Partially Blind RSA Signatures
  • Benchmarks
  • Current Status
Background: Blind RSA Signatures

msg = “new_attendee”

Key = Pk, Sk
Background: Blind RSA Signatures

Blind_sig = BlindSign(Sk, Blinded_msg)

Blinded_msg, inv = Blind (Pk, msg)

Client

msg, Pk

Blinded_msg

Server

Pk, Sk

Blind_sig = BlindSign(Sk, Blinded_msg)
Background: Blind RSA Signatures

Signature = Finalize( Pk, msg, inv, Blind_sig )
Background: Blind RSA Signatures

• Input message is encoded before being blinded
  • PSS Encoding
• Signature is verified as a sub-routine in Finalize
• Signature is publicly verifiable!
Outline

• Motivation
• Background: Blind RSA Signatures
• Partially Blind RSA Signatures
• Benchmarks
• Current Status
Partially Blind RSA Signatures [1,2,3]

- Use same public metadata (md) needed in all stages
  - Blinding
  - Signing
  - Finalizing
  - Verifying

3. Amjad, G., Yeo, K., Yung, M. RSA Blind Signatures with Public Metadata (in submission).
Partially Blind RSA Signatures

• Augmented Input Message
  • Unique encoding of message and “md” passed to PSS encoding

• Derive the Public Key based on public metadata
  • H(md)
    • using HKDF as H for implementation ease
  • H(md) needs to be co-prime to $\phi(N)$ where N is the RSA modulus
    • Strong RSA modulus (should be a product of two safe primes)
    • H(md) output is size restricted and an odd number
Security Considerations

- One-more-unforgeability
- Unlinkability under same public metadata
- Domain separation
  - Different RSA moduli will ensure different derived public keys for same public metadata
  - Hash functions in input message augmentation and public key derivation are domain separated
- Denial of service
Outline

• Motivation
• Background: Blind RSA Signatures
• Partially Blind RSA Signatures
• Benchmarks
• Current Status
# Benchmarks

<table>
<thead>
<tr>
<th></th>
<th>Blind RSA Signatures</th>
<th>Partially Blind RSA Signatures</th>
</tr>
</thead>
<tbody>
<tr>
<td>Blind</td>
<td>0.31 ms</td>
<td>1.4 ms</td>
</tr>
<tr>
<td>BlindSign</td>
<td>1.6 ms</td>
<td>4.3 ms</td>
</tr>
<tr>
<td>Finalize</td>
<td>0.028 ms</td>
<td>1.1 ms</td>
</tr>
</tbody>
</table>

* [https://github.com/google/anonymous-tokens](https://github.com/google/anonymous-tokens)
Outline

• Motivation
• Background: Blind RSA Signatures
• Partially Blind RSA Signatures
• Benchmarks
• Current Status
Current Status

- At Google:
  - Chrome IP Protection Authentication
  - Pixel VPN Authentication

- Third-party audit of the cryptography

- Academic paper with accompanying security proofs / analysis is currently in submission
  - Updated IACR e-print will be available by end of year

- Interest in adopting this document?
Thank you!