On the Multi-User Security of Short Schnorr Signatures with Preprocessing

Jeremiah Blocki  Seunghoon Lee
Department of Computer Science, Purdue University

Summary

- Schnorr Signatures: 4k bits long with k bits of security
- **Short Schnorr Signatures**: 3k bits long (truncating hash output)

Questions:

- k bits of multi-user security for short Schnorr signatures?
- Is the short Schnorr signatures secure against preprocessing attacks?

Our Result:

- Single/Multi-user security of short Schnorr signatures (in GGM+ROM)
- Multi-User Security of short Schnorr signatures against preprocessing
- Applicable to other Fiat-Shamir-based signatures

Schnorr Signature Scheme [2]

\[
\begin{align*}
\text{Kg}(1^k) : & \quad \text{Sign}(sk, m) : \\
& \text{sk} \leftarrow \mathbb{Z}_p \quad \text{pk} \leftarrow g^\text{sk} \\
& \text{return} (pk, sk) \\
Vfy(pk, m, c) : & \quad \text{Parse} \ (a, c) = H(m) \\
& \text{if} \ H(R|\text{msg}) = c \ \text{then} \\
& \text{return} 1 \\
& \text{else} \text{return} 0 \\
\end{align*}
\]

Our Results in Detail

**Multi-User Security of Short Schnorr Signatures**

- Theorem (Informal). Given N public keys, any attacker making \(q\) queries can forge a short Schnorr signature with probability \(O(2^Nq^2)\) in the GGM (of order \(p \geq 2^{2k}\)) plus ROM.

- If \(k = 112\) (i.e., \(p = 2^{224}\)) and \(N = 2^{32}\) (more than the half of the entire world population), an attacker making at most \(q = 2^{60}\) queries succeeds with probability \(\frac{1}{2} \approx 2^{-30}\).
- A naive reduction loses a factor of \(N\), i.e., \(\frac{1}{2N} \approx 2^{-30}\).

**Multi-User Security of Short Schnorr Signatures against Preprocessing**

- Theorem (Informal). Given N public keys, any **preprocessing** attacker making \(q_{\text{pre}}\) queries and outputs an s-bit hint (preprocessing phase) and making \(q_{\text{on}}\) queries (online phase) can forge a key-prefixed short Schnorr signature with probability \(O(s^2 N + 2^k + q_{\text{on}}^2)\) in the GGM (of order \(p > 2^{2k}\)) plus ROM.

- Why key-prefixing? ✗ Not to allow \(c = 0\) signatures!
- Setting \(p = 2^k N^2\) and \(S = 2^{1/2}\) and \(N = 2^{N/4}\) signature length: \(k + \log_2 p \approx 3.75k\).
- Still achieving \(k\) bits of multi-user security!

Similar bounds are applicable to other Fiat-Shamir-based signatures, i.e., key-prefixed Chaum-Pedersen-FDH signatures [5] and short Katz-Wang signatures [6]

- Katz-Wang signature length: 4k bits
- Our short Katz-Wang signature length: \(3k + \log_2 N + \log_2 S + \log_2 (2k + \log_2 NS)\) bits (for preprocessing)

Our Techniques

- **Restricted Discrete-Log Oracle**: We consider a stronger attacker who is given access to a restricted discrete-log oracle \(\text{DLog}(\cdot)\)

  \[
  \begin{align*}
  & \tau_1, \tau_2, \ldots, \tau_k \text{ are publicly given!} \\
  & \tau(x_1 + 3x_2 - 5) \not\in \text{DLog}() \quad \text{not fresh} \\
  & \tau(x_1) \quad \text{fresh} \\
  & \tau(y) \quad \text{fresh} \\
  & \tau(y_0) \quad \text{fresh} \quad \text{why restricted? ✗ To rule out trivial attacks!}
  \end{align*}
  \]

  **Security Reduction**

  - Bridge inputs \(\tau(x_1), \ldots, \tau(x_N)\) are public signing keys when simulating \(A_{\text{sec}}\).
  - The reduction also make use of a programmable random oracle whenever \(A_{\text{sec}}\) queries \(\text{Sign}(\cdot)\) for a particular user \(j \in [N]\).
  - Probability of failure events is negligible: \(\Pr[A_{\text{sec}} \Rightarrow \text{fail}] \leq \Pr[A_{\text{sec}} \Rightarrow \text{fail}] + \Pr[\text{fail}] \leq O(2^{-q_{\text{on}}})\).

References