Fortgeschrittene Techniken der Kryptographie

Jürgen Fuß

Episode 12: Stateful Signatures und Stateless Signatures

Stateful Signatures

Merkle Signatures – Schlüsselerzeugung

- Es werden 2^{32} Lamport-Schlüsselpaare erzeugt.
- Die Lamport Private Keys werden als **Signaturschlüssel** verwendet.
- Die Lamport Public Keys sind die Datensätze für einen Merkle Tree.
- Die Merkle Root ist der **Public Key**, der zum Verifizieren von Signaturen verwendet wird.
- Anstatt die Private Keys zufällig zu erzeugen, werden sie mit einem PRNG aus einem zufällig gewählten Seed generiert. So können sie auch nach der Erzeugung der Merkle Root wieder gelöscht und für das Signieren aus dem Seed mit dem PRNG neu berechnet werden.

Merkle Signatures – Signieren

- Für das Signieren muss sichergestellt werden, dass keiner der Lamport Private Keys ein zweites Mal verwendet wird. Eine Möglichkeit ist, die Private Keys der Reihe nach zu verwenden. Dann braucht bloß ein Zähler als **State** gespeichert zu werden.
- Das Signieren läuft wie beim Lamport-Verfahren ab.
- Zusätzlich wird der für die Verifikation zu verwendende Lamport Public Key an die Signatur angehängt, denn für die Verifikation liegt nur die Merkle Root vor.
- Zusätzlich wird für diesen Datensatz im Merkle Tree auch noch der Authentication Path für den Verifikationsschlüssel an die Signatur angehängt.

Merkle Signatures – Verifizieren

- Für die Verifikation wird zunächst die Lamport-Signatur mit dem mitgesendeten Lamport Public Key geprüft.
- Dann wird mit diesem Public Key und dem Authentication Path gegen die Merkle Root geprüft, die den Signatur-Public-Key bildet.

XMSS (eXtended Merkle Signature Scheme, RFC 8391)

Im XMSS-Verfahren (eXtended Merkle Signature Scheme, RFC 8391) konnte die Länge von Signaturen weiter verkleinert werden.
Als One-Time-Signaturverfahren wird eine verbesserte Version des Winternitz-Verfahrens, das sog. WOTS+-Verfahren als One-Time-Signaturverfahren eingesetzt; das Verfahren wird so modifiziert, dass anstatt der Kollisionsresistenz der Hashfunktion nur deren schwache Kollisionsresistenz benötigt wird.

XMSS^{MT} (XMSS Multitree)

Die Idee von Goldreich ist, zunächst einen Merkle Tree der Tiefe 16 für 2^{16} One-Time Public Keys zu erstellen. Dann werden die ersten 2^{16} eigentlichen One-Time-Schlüsselpaare erstellt und für die 2^{16} Public Keys wie im Merkle-Verfahren die Merkle Root berechnet. Diese Merkle Root wird mit dem ersten Schlüssel aus dem ersten Baum signiert.

XMSS^MT (2)

Beim Signieren wird wie im Merkle-Verfahren vorgegangen. Als Signatur werden die One-Time-Signatur der zu signierenden Daten, der Authentication Path und die Merkle Root des verwendeten (unteren) Trees, sowie die One-Time-Signatur über dessen (untere) Merkle Root und der Authentication Path für den (oberen) Merkle Tree verwendet.
Die Prüfung solcher Signaturen läuft wie die Prüfung einer Zertifikatskette ab. Zunächst kann überprüft werden, dass die Merkle Root korrekt ist. Dann kann mit dieser Merkle Root die Signatur über die Daten (wie im Merkle-Verfahren) überprüft werden.

XMSS^MT (3)

Der große Vorteil liegt darin, dass anstatt mit einem großen Merkle Tree mit zwei wesentlich kleineren Merkle Trees gearbeitet werden kann. Die kleineren Bäume haben um den Faktor $2^{16} = 65536$ weniger Blätter, was alle Berechnungen wesentlich schneller macht.
Sobald 2^{16} Signaturen erstellt worden sind, ist der erste (untere) Baum aufgebraucht. Nun wird der zweite Public Key aus dem oberen Baum verwendet, um die Merkle Root eines zweiten (unteren) Baums zu signieren.
Insgesamt stehen $2^{16} \cdot 2^{16}$, also wieder 2^{32}, Signaturschlüsselpaare zu Verfügung.
Diese Idee kann auch mit mehr als zwei Ebenen von Bäumen umgesetzt werden. Das XMSS^{MT} (eXtended Merkle Signature Scheme with Multiple Trees, RFC 8391) verwendet bspw. vier Ebenen jeweils mit Bäumen der Tiefe 20. Damit werden 2^{80} Signaturen möglich, im Grunde also eine unbegrenzte Menge an Signaturen. Dennoch muss stets nur mit vier Bäumen mit jeweils 2^{20} Blättern gearbeitet werden.

HORS – Schlüsselerzeugung

- Wähle eine Hashfunktion h mit 256 Bit Outputlänge.
- Der **Private Key** Pr besteht aus 2^{16} zufällig gewählten Bitfolgen

$$s_0, \ldots, s_{65535}$$

  der Länge 256.
- Der dazugehörige **Public Key** besteht aus den Hashwerten

$$p_0 := h(s_0), \ldots, p_{65535} := h(s_{65535}).$$

- Public und Private Keys sind 2 MiB groß.

HORS – Signieren und Verifizieren

- Um einen 256 Bit langen (Hash-)Wert m zu signieren, wird er in 32 16-Bit-Blöcke $m_0, \ldots, m_{31}$ zerlegt. Diese werden als Zahlen zwischen 0 und 65535 interpretiert.
- Die **Signatur** von m ist dann

$$\mathrm{Sign}_{Pr}(m) = (s_{m_0}, \ldots, s_{m_{31}}).$$

  Signaturen sind somit 1 KiB groß.

- Zur **Verifikation** wird jeder 256-Bit-Block der Signatur gehasht und die Ergebnisse mit den entsprechenden Teilen des Public Key verglichen.

Few-Time Signatures und HORST

- Ein Problem mit der Unfälschbarkeit ergibt sich, wenn ein Schlüssel zum Signieren von zwei Hashwerten verwendet wird, bei denen gleiche 16-Bit-Blöcke vorkommen.
- Die Parameter sind hier so gewählt, dass dies bei weniger als 128 Nachrichten mit einer Wahrscheinlichkeit von unter 2^{-128} passiert. Solange also wenige Werte signiert werden, treten praktisch keine Kollisionen auf (**Few-Time Signature**).

- Das HORS-Verfahren kann zum HORST-Verfahren (HORS with Trees) erweitert werden, indem ein Merkle Tree der Tiefe 16 für die 2^{16} Public Keys verwendet wird.

SPHINCS+
(Stateless Practical Hash-based Incredibly Nice Cryptographic Signatures)

- Zum Signieren einer Nachricht wird das HORST-Few-Time-Signaturverfahren eingesetzt.
- Der Schlüssel zum Signieren einer Nachricht wird wird zufällig ausgewählt.
- So kommen Schlüssel u.U. auch öfter zum Einsatz, was aber bei dem Few-Time-Signaturverfahren kein Problem mehr darstellt.
- Public Keys werden wieder in Merkle Trees gesammelt.
- Wie bei XMSS^{MT} wird wieder eine Kaskade von Merkle Trees mit WOTS+-Signaturen über Merkle Roots verwendet.

SPHINCS+ gehört zu den ersten Post-Quantum-Signaturverfahren, die 2022 vom NIST zur Standardisierung ausgewählt wurden.