Design des Secret Sharings

Problem

Es wurde festgestellt, dass es nicht sinnvoll ist auf den Secrets eine Operation zu erlauben wo *beliebige* Teilsummen aufaddiert werden können. Denn dies könnte dazu ausgenutzt werden, unabhängig davon welcher Sharing Mechanismus benutzt würde, die Secrets zu rekonstruieren. Daher soll bei dem neuen Design eine Berechnung beliebiger Teilsummern nicht möglich sein. Jetzt gilt es Anwendungsfälle zu finden wo nicht das Vorhandensein von Wissen über einzelne Secrets notwendig ist.

Ideen

- Die Secrets könnten also in Blöcken zusammengefasst werden zu einem neuen **Blocksecret** für welches dann die Shares erzeugt werden.
- Die **Blocksecrets** werden durch einen Zufallswert r_i „linear verschoben“ der die Aufdeckung erschwert: $s' \coloneqq s + r$.

Fragen

Wieso sprechen wir im Arbeitstitel von einer „Integritätsprüfung homomorpher Operationen“, da es doch eher darum geht das die homomorphe Operation vertraulich arbeitet, also beim Rechnen keine Information leakt.

Anmerkungen

Die Durchführung teilt sich grundsätzlich in zwei Teile:

- Das Secret Sharing, also das Aufteilen eines Secrets und dessen Rekonstruktion. Hierdurch soll Malleability in der homomorphen Operation vermieden werden.
- Eine homomorphe Operation auf den Secrets zu Analysezwecken. Hier: Addition

Umsetzung von Blocksecrets

Definition n-pair-cover 1. *Ein n-pair-cover erreicht bedingte Vertraulichkeit von secrets $s_1, \ldots, s_n$ in dem nur ein Zugriff auf ihre Summe ermöglicht wird. Dadurch verdecken sich die Secret gegenseitig, da eine Aufdeckung der Summe verschiedene Paarkombinationen zulässt.*

Beispiel 2-pair-cover

Im Falle von $n = 2$ hat die Aufdeckung des Supersecrets $\mathbb{N} \ni s_{i,j} = s_i + s_j$ die Paarkombinationen $(1, s_{i,j} - 1), (2, s_{i,j} - 2), \ldots, (\lfloor \frac{n}{2} \rfloor, \lfloor \frac{n+1}{2} \rfloor)$.
Je größer $s_{i,j}$, desto mehr Paarkombinationen gibt es und für einen Angreifer sinkt die Aufdeckungswahrscheinlichkeit des Paars gegen 0: $\lim_{n\to\infty} \frac{1}{\lfloor \frac{n}{2} \rfloor} = 0$

Algorithmus 2-pair-cover

Gegeben: Secrets $s_1, \ldots, s_n$

1. Falls n ungerade, füge $s_{n+1} = \text{rand}(\max(s_1, \ldots, s_n)$ hinzu. Hierdurch wird gewährleistet, dass für das letzte Secret auch ein Supersecret erzeugt werden kann.
2. Wähle *zufällige* Indices $i, j \in 1, \ldots, n$. Der Grund für die zufällige Wahl der Indices gründet sich in der Tatsache, dass auf den Secret ein Struktur vorhanden sein könnte (z.B. aufsteigende Zahlen). Würde man nun ein Supersecret aus benachbarten Secrets erzeugen, so würde die Aufdeckung des Supersecrets die Wahl möglicher Paarkombinationen einschränken da davon auszugehen ist, dass s_i und s_j relativ nahe beieinander liegen.
3. Erzeuge das Supersecret $s_{i,j} \coloneqq s_i + s_j$
4. Lösche s_i und s_j
5. Wiederhole Schritte $2 - 4$ solange es noch Secrets mit nur einem Index gibt.

Ausgabe: Bedingt zufällig bestimmte Supersecrets $s_{i_1,j_1}, \ldots, s_{i_{\lfloor \frac{n}{2} \rfloor}, j_{\lfloor \frac{n+1}{2} \rfloor}}$

Anmerkungen

Es ist weiter möglich beliebige Summen über die Supersecrets zu berechnen. Für statistische Zwecke können auch Mittelwerte oder Varianzen bestimmt werden. Die Summen, Mittelwerte und Varianzen entsprechen dann den echten Summen, Mittelwerten und Varianzen der Secrets aus welchen die Supersecrets erzeugt wurden.
Warum wurde eine lineare Verschiebung mit Zufallswerten nicht näher in Erwägung gezogen? Es wurde angenommen, dass der Analyzer nur mit unverfälschten Supersecrets arbeiten will. Sind die Werte verschoben, so wären die Analysen weniger aussagekräftig.