Combinatorial Methods in the Context of Privacy

Browser Fingerprinting

- Browsers are widely used to consume services over the Internet.
- Transport Layer Security (TLS) is used to keep connection secure.
- Different TLS-Implementations expose different behaviour, especially when exposed to manipulated Handshakes.
- This allows distinction between browsers.

**Testing Framework**

- **Browser**: Client Hello
- **Testing Framework**: Finished
  - Change Cipher Spec Message
  - Server Hello Done
  - Server Key Exchange

**Sequences**

- The six server-side TLS handshake messages are regarded as a set of six abstract events $\mathcal{M}$.
- $\mathcal{M} = \{\text{ServerHello}, \text{Cert}, \text{ServerKeyExchange}, \text{ServerHelloDone}, \text{ChangeCipherSpec}, \text{Finished}\}$.
- All permutations of any non-empty subset $\mathcal{E}$ of $\mathcal{M}$ are tested, therefore all SCAs defined over the elements $\mathcal{E}$ are included.
- A row in a SCA created in such a way is a test sequence that can be transformed to a sequence of actual TLS handshake messages.
- Example: $\{0, 3, 2, 1\}$ translates to $\{\text{ServerHello}, \text{ServerHelloDone}, \text{ServerKeyExchange}, \text{Cert}\}$.

**Testing Methodology**

- Java-based software that automatically records behaviour of Browsers
- Uses TLS-Attacker to execute manipulated TLS Handshakes
- Only the sequence of the six server-side TLS messages is altered
- All possible $\sum_{k=1}^{6} \binom{6}{k} \cdot k! = 1956$ permutations were tested

**Decision Tree Based on a Single Test**

- Five Browsers were tested (Chrome, IE, Edge, Firefox and Opera).
- The best possible splitting was based on the browser families: {Chrome, Opera} {IE, Edge} {Firefox}.
- Distinguished by pairwise comparison of their feature vector.
- The best possible distinction was achieved even using sequences of length 2 (higher-strength sequences also yielded this outcome).