Distance Hijacking Attacks on Distance Bounding Protocols

Cas Cremers
ETH Zurich

Joint work with:
Kasper Rasmussen, Benedikt Schmidt, Srdjan Capkun
Distance Bounding
Distance Bounding Protocols

- Objective: **ensure proximity**

- Protocol with two roles: **Prover** and **Verifier**

- **Verifier obtains an upper bound** on the distance to the prover

- Guarantee also holds if the prover is malicious
Distance bounding for network access
Brands and Chaum protocol (1993)

**Phase 1:**

**Setup**

- **Verifier**
  - Fresh $nv$
  - Commit($np$)

- **Prover**
  - Fresh $np$

**Phase 2:**

**Fast response phase**

- Measure response time
- $nv$
- $nv \oplus np$

**Phase 3:**

**Finalize**

- Verify commit and signature
- $np, \text{sign}(P, <nv, nv \oplus np>)$
Threats considered in protocol proposals

Mafia Fraud

- **External attacker** modifies distance of **honest prover**

Distance Fraud

- **Dishonest prover** modifies his own distance

Terrorist Fraud

- **Dishonest prover** collaborates with **closer attacker** to modify his distance
What about other honest provers?
Distance Hijacking attack on B&C

Phase 2: Fast response phase

Verify commit and signature

Measure response time

Dishonest P

np, sign(P, <nv, nv xor np>)

fresh np

Honest P'

nv xor np

nv

commit(np)

fresh nv

V

Distance Hijacking

A **Distance Hijacking attack** is an attack in which a **dishonest prover** $P$ **exploits** one or more **honest parties** to provide a verifier $V$ with false information about the distance between $P$ and $V$. 

![Diagram](image)
## Scope

About half of the investigated protocols vulnerable

- Brands and Chaum based designs usually vulnerable
- Hancke & Kuhn based designs seem okay

<table>
<thead>
<tr>
<th>Protocol</th>
<th>DH-attack?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Brands and Chaum (Fiat-Shamir)</td>
<td>Yes</td>
</tr>
<tr>
<td>Brands and Chaum (Schnorr)</td>
<td>Yes</td>
</tr>
<tr>
<td>Brands and Chaum (signature)</td>
<td>Yes</td>
</tr>
<tr>
<td>Bussard and Bagga</td>
<td>-</td>
</tr>
<tr>
<td>CRCS</td>
<td>Yes</td>
</tr>
<tr>
<td>Hancke and Kuhn</td>
<td>-</td>
</tr>
<tr>
<td>Hitomi</td>
<td>-</td>
</tr>
<tr>
<td>KA2</td>
<td>-</td>
</tr>
<tr>
<td>Kuhn, Luecken, Tippenhauer</td>
<td>Yes</td>
</tr>
<tr>
<td>MAD</td>
<td>Yes</td>
</tr>
<tr>
<td>Meadows et al for F(\ldots) = \langle NV, NP xor P \rangle</td>
<td>Yes</td>
</tr>
<tr>
<td>Munilla and Peinado</td>
<td>-</td>
</tr>
<tr>
<td>Noise resilient MAD</td>
<td>Yes</td>
</tr>
<tr>
<td>Poulidor</td>
<td>-</td>
</tr>
<tr>
<td>Reid et al.</td>
<td>-</td>
</tr>
<tr>
<td>Swiss-knife</td>
<td>-</td>
</tr>
<tr>
<td>Tree</td>
<td>-</td>
</tr>
<tr>
<td>WSBBC+DB</td>
<td>Yes</td>
</tr>
<tr>
<td>WSBBC+DB Noent</td>
<td>Yes</td>
</tr>
</tbody>
</table>
Fixing the problem

- **Secure channel** (TLS) does not help here
  - Cannot use cryptography during fast response
  - Protocols that use secure channels in the other phases may still be vulnerable

- Fixes **logically bind fast response** to other phases
  - Involve identity in response
  - Bind identity to nonce in Phase 1
  - Fixes do not require additional cryptography
Formal model

- We extended Basin et al. [TPHOLs'09]

**Hybrid symbolic model**
- Also captures bit-level overshadowing attacks
  - adversary flips some bits of an unknown message
- Formalization in Isabelle/HOL

- Used to show that our fixes prevent the found attacks
  (Details in the paper; theory files publicly available)
Multiple protocols

Interaction between protocols with similar fast response hardware can lead to attacks

- Similar to "chosen protocol" or "multi-protocol" attacks
- ALL protocols vulnerable
Are all attacks now covered?

- Distance Fraud
- Mafia Fraud
- Terrorist Fraud
- Distance Hijacking
Restructuring attacks on DB protocols

Assume an attack trace where V computes incorrect distance for P

A **Distance Hijacking attack** is an attack in which a dishonest prover P exploits one or more honest parties to provide a verifier V with false information about the distance between P and V.
Conclusions

- Many protocols vulnerable to **Distance Hijacking**
  - Fixes do not introduce significant overhead
  - Just-in-time: distance bounding implementations starting to be produced

- Distance Hijacking is a **relevant threat** in many cases

- Cannot afford to ignore multiple provers/verifiers during analysis

- Interaction between different DB-protocols still possible...