On Key-collisions in (EC)DSA Schemes

CRYPTO 2002 Rump Session

Tomáš Rosa
tomas.rosa@i.cz,
http://crypto.hyperlink.cz
ICZ, a.s.
Dept. of Computer Science, CTU in Prague
CZECH REPUBLIC
On Key-collisions in (EC)DSA Schemes (1)

- Let \((m, S)\) be a message and its signature.
- Let us have two different public keys \((Pub_A, Pub_B)\), such that:
  - \(\text{VER}_{Pub_A}(m, S) = \text{VER}_{Pub_B}(m, S) = \text{VALID\_SIGNATURE}\).
- Then \((Pub_A, Pub_B)\) is said to be a key-collision (k-collision).
- The signature \(S\) is referred to as a \(k\)-colliding signature.
An ability to find a $k$-collision for an arbitrary $(m, S)$ may lead to attacks on a non-repudiation service.

- Leads to: “It has been somebody else, who has signed that message...”

There are also non-cooperatively computable $k$-collisions.

- Leads to: “It has been me, who has signed that message, not her/him...”
Non-cooperatively computable $k$-collisions are trivially feasible in DSA for an arbitrary $(m, S)$ and $Pub_A$.

The algorithm uses a partial inversion of the DSA instance generation process.
- It exploits the lack of restrictions on the value of the subgroup generator $g$.

Due to common algebraic properties this attack easily extends on ECDSA too.
Countermeasures

- **Main:** Fix the FIPS 186-2, or make own proprietary extensions; the value of \( g \) should be associated with a certificate of its proper generation.
- **Temporary:** Include detailed public key information into the data to be signed.
  - Must be done carefully and with respect to a particular PKI protocol.
  - Still vulnerable through a 2\(^{nd}\) order \( k \)-collision: different messages, different keys, the same signature.

Tomáš Rosa, tomas.rosa@i.cz