Transcript Collision Attacks: Breaking Authentication in TLS, IKE and SSH

or: MD5 MUST DIE

http://sloth-attack.org

Karthikeyan Bhargavan
Gaëtan Leurent
Agility vs. Downgrade Attacks

Crypto protocols and applications evolve

- SSL v3 ➔ TLS 1.2
- DH-768 ➔ Curve25519
- MD5 ➔ SHA-256

Agility: graceful transition from old to new
- Negotiate best shared version, cipher, DH group

What can go wrong?
- We get lazy and forget to remove weak algorithms
- Downgrade attacks: POODLE, LOGJAM, SLOTH
Authenticated DH with Negotiation

\[ A \]

Knows \( s_{k_A}, p_{k_B}, (g, p) \)

\[ B \]

Knows \( s_{k_B}, p_{k_A}, (g, p) \)

\[ m_1 = g^x \mod p | params_A \]

\[ m_2 = g^y \mod p | params_B \]

Keys \( k = \text{kdf}(g^{xy} \mod p) \)

Transcript \( t = T(m_1, m_2) \)

\[ \text{sign}(s_{k_A}, \text{hash}(t)), \text{mac}(k, A) \]

\[ \text{sign}(s_{k_B}, \text{hash}(t)), \text{mac}(k, B) \]

Signed Transcript
What Transcript to Sign?

• Sign the full message trace
  – \texttt{sign}(sk_B, \texttt{hash}(m_1 \mid m_2))
  – \textit{Example}: TLS 1.3, SSH-2, TLS 1.2 client auth

• Sign your ephemerals, MAC the transcript
  – \texttt{sign}(sk_B, \texttt{hash}(\texttt{nonce}_A \mid \texttt{nonce}_B \mid g \mid p \mid g^y))
  – \textit{Example}: TLS 1.2 server auth

• Sign your own messages and MACed identity
  – \texttt{sign}(sk_A, \texttt{hash}(m_1 \mid \texttt{mac}(k,A)))
  – \texttt{sign}(sk_B, \texttt{hash}(m_2 \mid \texttt{mac}(k,B)))
  – \textit{Example}: IKEv2 initiator, responder, EAP auth
Using Weak Hash Functions

• Sign the full transcript
  – \texttt{sign}(sk_B, \texttt{hash}(m_1 \mid m_2))
  – \textit{Example}: TLS 1.3, SSH-2, TLS 1.2 client auth

• How weak can the \texttt{hash} function be?
  – do we need collision resistance?
  – do we only need 2\textsuperscript{nd} preimage resistance?
  – Is it still safe to use MD5, SHA-1 in TLS, IKE, SSH?
  – \textit{Disagreement}: cryptographers vs. practitioners (see Schneier vs. Hoffman, RFC4270)
SLOTH: Transcript Collision Attacks

**Man-in-the-Middle:**
network attacker/malicious server

- Parameter Downgrade
- Keys $k_a = \text{kdf}(g^{xy'} \mod p)$
- Transcript $t_a = T(m_1, m_2')$
- $m_1 = g^x \mid \text{params}_A$
- $m_2' = g^{y'} \mid \text{params}'_B$

**Collision** $\text{hash}(t_a) = \text{hash}(t_b)$

- Keys $k_b = \text{kdf}(g^{x'y} \mod p)$
- Transcript $t_b = T(m_1', m_2)$

**Server Impersonation**
- $\text{sign}(sk_A, \text{hash}(t_b)), \text{mac}(k, A)$
- $\text{sign}(sk_B, \text{hash}(t_b)), \text{mac}(k, B)$

**Client Impersonation**
- $\text{sign}(sk_A, \text{hash}(t_b)), \text{mac}(k, A)$
- $\text{sign}(sk_B, \text{hash}(t_b)), \text{mac}(k, B)$

- A knows $sk_A, pk_B$
- MitM knows $sk_B, pk_A$
Computing a Transcript Collision

\[
\text{hash}(m_1 | m'_2) = \text{hash}(m'_1 | m_2)
\]

- We need to compute a collision, *not a preimage*
  - Attacker controls parts of both transcripts
  - If we know the black bits, can we compute the red bits?
  - This is usually called a *generic collision*

- If we’re lucky, we can set up a *shortcut* collision
  - **Common-prefix**: collision after a shared transcript prefix
  - **Chosen-prefix**: collision after attacker-controlled prefixes
Primer on Hash Collision Complexity

• MD5: known attack complexities
  – **MD5** second preimage: $2^{128}$ hashes
  – **MD5** generic collision: $2^{64}$ hashes
    (birthday)
  – **MD5** chosen-prefix collision: $2^{39}$ hashes (1 hour)
  – **MD5** common-prefix collision: $2^{16}$ hashes (seconds)

• SHA1: estimated attack complexities
  – **SHA1** second preimage: $2^{160}$ hashes
  – **SHA1** generic collision: $2^{80}$ hashes
    (birthday)
  – **SHA1** chosen-prefix collision: $2^{77}$ hashes (2)
Composite Hash Constructions

- When used as transcript hash functions, many constructions are not collision resistant
  - $\text{MD5}(x) \ | \ \text{SHA1}(x)$
    not much better than SHA1
  - $\text{HMAC-MD5}(k,x)$
    not much better than MD5
  - $\text{HMAC-SHA256}(k,\text{MD5}(x))$
    not much better than MD5
  - $\text{Truncated HMAC-SHA256}(k,x)$ to N bits
    not much better than a N bit hash function
Computing Transcript Collisions

A

hash

\[ \text{len}_1 \]
\[ g^x \]
\[ \text{params}_A \]

\[ \text{len}_2' \]
\[ g^{y'} \]
\[ \text{params}'_B \]

MitM

\[ m_1 \]
\[ m_1' \]

B

hash

\[ \text{len}_1' \]
\[ g^{x'} \]
\[ \text{params}'_A \]

\[ \text{len}_2 \]
\[ g^y \]
\[ \text{params}_B \]
Generic Transcript Collisions

A

<table>
<thead>
<tr>
<th>hash</th>
</tr>
</thead>
<tbody>
<tr>
<td>len₁</td>
</tr>
<tr>
<td>g^x</td>
</tr>
<tr>
<td>nonceₐ</td>
</tr>
</tbody>
</table>

B

<table>
<thead>
<tr>
<th>hash</th>
</tr>
</thead>
<tbody>
<tr>
<td>len₁'</td>
</tr>
<tr>
<td>g^x'</td>
</tr>
<tr>
<td>nonceₙ</td>
</tr>
</tbody>
</table>

MitM

Try random nonces until collision

N = 2^|hash|/2

MD5: 2⁶⁴
SHA-1: 2⁸⁰
HMAC/96: 2⁴⁸

Pre-emptible:
Static DH key, no fresh nonce

Try random nonces until collision

N = 2^|hash|/2

MD5: 2⁶⁴
SHA-1: 2⁸⁰
HMAC/96: 2⁴⁸
Chosen-Prefix Transcript Collisions

A

\[ \text{len}_1 \]
\[ g^x \]
\[ \text{blob}_A \]

MitM

B

\[ \text{len}_2 \]
\[ g^y \]
\[ \text{blob}_B \]

Known length, ephemeral DH key, arbitrary BLOB
Find Chosen-Prefix Collision $C_1, C_2$

$N = 2^{\text{CPC(hash)}}$

- MD5: $2^{39}$
- SHA-1: $2^{77}$
- HMAC/96: n/a
SLOTH: Attacking TLS 1.2 Client Auth

• TLS 1.2 upgraded hash functions used in TLS
  – SHA-256 for all handshake constructions
  – New signature algorithms extension: SHA-256/384/512

• TLS 1.2 added support for MD5-based signatures!
  – Even if the client and server prefer RSA-SHA256, the connection can be *downgraded* to RSA-MD5!

• Transcript collisions break TLS 1.2 client signatures
  – Chosen prefix collision attack using flexible formats
  – Demo: Takes 1 hour/connection on a 48-core workstation
  – Not very practical: connection must be live during attack
SLOTH: Attacking TLS Server Auth

- TLS 1.2 server signatures are harder to break
  - *Irony:* the weakness that enables Logjam blocks SLOTH
  - Needs $2^x$ prior connections + $2^{128-x}$ hashes/connection
  - Not practical for academics, as far as we know

- TLS 1.3 server signatures is potentially vulnerable
  - *New:* MD5, SHA-1 sigs now explicitly forbidden in TLS 1.3
Other SLOTH Vulnerabilities

• Reduced security for TLS 1.*, IKEv1, IKEv2, SSH
  – Impersonation attack on TLS channel bindings
  – Exploit downgrades + transcript collisions
  – These are protocol flaws, not implementation bugs
  – Main mitigation is to disable weak hash functions

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Property</th>
<th>Mechanism</th>
<th>Attack</th>
<th>Collision Type</th>
<th>Precomp.</th>
<th>Work/conn.</th>
<th>Preimage</th>
<th>Wall-clock time</th>
</tr>
</thead>
<tbody>
<tr>
<td>TLS 1.2</td>
<td>Client Auth</td>
<td>RSA-MD5</td>
<td>Impersonation</td>
<td>Chosen Prefix</td>
<td></td>
<td>2^{39}</td>
<td>2^{128}</td>
<td>48 core hours</td>
</tr>
<tr>
<td>TLS 1.3</td>
<td>Server Auth</td>
<td>RSA-MD5</td>
<td>Impersonation</td>
<td>Chosen Prefix</td>
<td></td>
<td>2^{39}</td>
<td>2^{128}</td>
<td>48 core hours</td>
</tr>
<tr>
<td>TLS 1.0-1.2</td>
<td>Channel Binding</td>
<td>HMAC (96 bits)</td>
<td>Impersonation</td>
<td>Generic</td>
<td></td>
<td>2^{48}</td>
<td>2^{296}</td>
<td>80 GPU days</td>
</tr>
<tr>
<td>TLS 1.2</td>
<td>Server Auth</td>
<td>RSA-MD5</td>
<td>Impersonation</td>
<td>Generic</td>
<td></td>
<td>2^{77}</td>
<td>2^{128}</td>
<td></td>
</tr>
<tr>
<td>TLS 1.0-1.1</td>
<td>Handshake Integrity</td>
<td>MD5</td>
<td>SHA-1</td>
<td>Downgrade</td>
<td></td>
<td>2^{77}</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IKE v1</td>
<td>Initiator Auth</td>
<td>HMAC-MD5</td>
<td>Impersonation</td>
<td>Generic</td>
<td></td>
<td>2^{65}</td>
<td>2^{128}</td>
<td></td>
</tr>
<tr>
<td>IKE v2</td>
<td>Initiator Auth</td>
<td>RSA-SHA-1</td>
<td>Impersonation</td>
<td>Chosen Prefix</td>
<td></td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SSH-2</td>
<td>Exchange Integrity</td>
<td>SHA-1</td>
<td>Downgrade</td>
<td>Chosen Prefix</td>
<td></td>
<td>2^{77}</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Final Thoughts

• Legacy crypto is strangely hard to get rid of, but we have to keep trying to kill broken primitives (MD5 MUST DIE)

• Key exchanges in Internet protocols do rely on collision resistance, question anyone who tells you otherwise!

• **Future**: new downgrade resilient protocols, collision-resistant authentication mechanisms

• More details, papers, demos are at: [http://sloth-attack.org](http://sloth-attack.org)