Security issues from bad crypto

Gaëtan Leurent

Joint work with:
Karthikeyan Bhargavan

Inria

Journées pre-GDR sécurité
Secure channel (TLS)

- Crypto provides **secure communication** against an adversary

```
Hello
Hello, Public key, Certificate
Session key (encrypted with public key)
```

```
AES-CBC(m₁), HMAC-SHA1(m₁)
AES-CBC(m₂), HMAC-SHA1(m₂)
```

- **Handshake protocol**
  - Establish session key using **public key** crypto

- **Record protocol**
  - Exchange application data using **secret key** crypto
Security of cryptographic protocols

Classical approach

- Security of the protocol
  - Security proofs assuming security of cryptographic operations
- Security of the modes (HMAC, CBC, ...)
  - Security proofs (assuming security of the primitive)
- Security of the primitives (AES, SHA-1, RSA, ...)
  - Studied with cryptanalysis

Problem

- Ciphers with known weaknesses are used in practice
  - Proof doesn’t hold anymore, but attacks are not obvious...
  - How theoretical are the attacks?
Security of cryptographic protocols

Classical approach

- Security of the protocol
  - Security proofs assuming security of cryptographic operations
- Security of the modes (HMAC, CBC, ...)
  - Security proofs (assuming security of the primitive)
- Security of the primitives (AES, SHA-1, RSA, ...)
  - Studied with cryptanalysis

Problem

- Ciphers with known weaknesses are used in practice
  - Proof doesn’t hold anymore, but attacks are not obvious...
  - How theoretical are the attacks?
Cryptography and security

- Cryptography is an element to build a secure system
  - There can be security issues at every step
  - But we mostly know how to build good crypto...

- User
- Application
- Protocol
- Mode
- Primitive
- Implementation

- Fishing, weak passwords, password reuse, ...
- SSL stripping, bad cert checks, encryption only, ...
- Padding oracle, predictable IV (BEAST)
- CBC collisions (Sweet32), CTR malleability (To)
- MD5 collisions, RC4 bias
- Side channel, buffer overflow, bugs (SM)
**Cryptography and security**

- **Cryptography** is an element to build a secure system
  - There can be security issues at every step
  - But we mostly know how to build good crypto...

![Cryptography Pyramid]

- **User**: Fishing, weak passwords, password reuse, ...
- **Application**: SSL stripping, bad cert checks, encryption only, ...
- **Protocol**: Padding oracle, predictable IV (BEAST)
- **Mode**: CBC collisions (Sweet32), CTR malleability (To)
- **Primitive**: MD5 collisions, RC4 bias
- **Implementation**: Side channel, buffer overflow, bugs (SMAK)
What is an attack?

For cryptographers

- Define expected security
- Anything faster is an attack
  - Eg. faster than trying all keys

For users

- Define attacker means
- Anything doable is an attack
  - Eg. one year on a PC

Attacks only get better

AES-256 has a 256-bit key

- Related-key attack with $2^{100}$ ops.
- Not a practical threat

Blowfish-32 has a 32-bit key

- No attack faster than $2^{32}$
- Key-search takes minutes
What is an attack?

For cryptographers

- Define expected security
- Anything faster is an attack
  - Eg. faster than trying all keys

Attacks only get better

AES-256 has a 256-bit key

- Related-key attack with $2^{100}$ ops.
- Not a practical threat

Blowfish-32 has a 32-bit key

- No attack faster than $2^{32}$
- Key-search takes minutes

For users

- Define attacker means
- Anything doable is an attack
  - Eg. one year on a PC
## What is an attack?

<table>
<thead>
<tr>
<th>For cryptographers</th>
<th>For users</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Define <strong>expected security</strong></td>
<td>- Define <strong>attacker means</strong></td>
</tr>
</tbody>
</table>
| - Anything faster is an attack  
  - *Eg.* faster than trying all keys | - Anything doable is an attack  
  - *Eg.* one year on a PC |

*Attacks only get better*

<table>
<thead>
<tr>
<th>For cryptographers</th>
<th>For users</th>
</tr>
</thead>
</table>
| - Attack **primitive**  
  - If broken, **stop using it**  
  - Proof hypothesis broken | - Does it break real **protocols**?  
  - Migration is **expensive** |
# Cryptanalysis in theory and in practice

## Cryptanalysis of MD5

<table>
<thead>
<tr>
<th>Year</th>
<th>Attack Type</th>
<th>Exploit Year</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1993</td>
<td>Compression function attack</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2005</td>
<td>Collision attack</td>
<td>2007</td>
<td>Exploitable in APOP</td>
</tr>
<tr>
<td>2007</td>
<td>Free-start collision attack</td>
<td>2009</td>
<td>Exploitable for rogue CA</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2013</td>
<td>Exploited by Flame</td>
</tr>
</tbody>
</table>

## Cryptanalysis of RC4

<table>
<thead>
<tr>
<th>Year</th>
<th>Attack Type</th>
<th>Exploit Year</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2000</td>
<td>Biases in RC4 keystream</td>
<td>2013</td>
<td>Exploitable in TLS</td>
</tr>
<tr>
<td>2001</td>
<td>Related-key attack on RC4</td>
<td>2002</td>
<td>Exploitable in WEP</td>
</tr>
</tbody>
</table>

## This talk

- Leverage weakness of crypto algorithms to break protocols

---

Gaëtan Leurent (Inria)  
Security issues from bad crypto  
Journées pre-GDR sécurité
Cryptanalysis in theory and in practice

Cryptanalysis of MD5

<table>
<thead>
<tr>
<th>Year</th>
<th>Attack Type</th>
<th>Year</th>
<th>Exploitation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1993</td>
<td>Compression function attack</td>
<td>2007</td>
<td>Exploitable in APOP</td>
</tr>
<tr>
<td>2005</td>
<td>Collision attack</td>
<td>2007</td>
<td>Exploitable for rogue CA</td>
</tr>
<tr>
<td>2007</td>
<td>Free-start collision attack</td>
<td>2013</td>
<td>Exploited by Flame</td>
</tr>
</tbody>
</table>

Cryptanalysis of RC4

<table>
<thead>
<tr>
<th>Year</th>
<th>Attack Type</th>
<th>Year</th>
<th>Exploitation</th>
</tr>
</thead>
<tbody>
<tr>
<td>2000</td>
<td>Biases in RC4 keystream</td>
<td>2013</td>
<td>Exploitable in TLS</td>
</tr>
<tr>
<td>2001</td>
<td>Related-key attack on RC4</td>
<td>2002</td>
<td>Exploitable in WEP</td>
</tr>
</tbody>
</table>

This talk

- Leverage **weakness** of crypto algorithms to **break protocols**
Outline

Security and Cryptography

CBC Collision Attack

In Practice

MD5 Collisions

Breaking APOP

SLOTH Attack
<table>
<thead>
<tr>
<th>Security and Cryptography</th>
<th>CBC Collision Attack</th>
<th>In Practice</th>
<th>MD5 Collisions</th>
<th>APOP</th>
<th>SLOTH</th>
<th>Conclusion</th>
</tr>
</thead>
</table>

Outline

Security and Cryptography

**CBC Collision Attack**

In Practice

MD5 Collisions

Breaking APOP

SLOTH Attack
Block ciphers and Modes of operation

▶ A block cipher is a **family of permutations**
▶ It is used with a **mode of operation** : CBC, CTR, GCM, ...
  ▶ To deal with variable-length messages
  ▶ To include randomness
  ▶ Important example : CBC

\[ E_k \]

\[ m \rightarrow n \]

\[ k \]

\[ E \]

\[ c \]

\[ n \]

\[ m_0 \rightarrow m_1 \rightarrow m_2 \rightarrow m_3 \]

\[ E_k \]

\[ c_{-1} \rightarrow c_0 \rightarrow c_1 \rightarrow c_2 \rightarrow c_3 \]

\[ IV \]
A block cipher is a **family of permutations**

It is used with a **mode of operation** : CBC, CTR, GCM, ...

- To deal with variable-length messages
- To include randomness
- Important example : CBC
CBC collisions

- Well known collision attack against CBC

If $c_i = c_j$, then $c_{i-1} \oplus m_i = c_{j-1} \oplus m_j$

Ciphertext collision reveals the xor of two plaintext blocks
The birthday paradox

In a room with 23 people, there is a 50% chance that two of them share the same birthday.

Security of CBC

- CBC leaks plaintext after $2^{n/2}$ blocks encrypted with the same key
- Security of mode can be lower than security of cipher
Birthday paradox

The birthday paradox

- In a room with 23 people, there is a 50% chance that two of them share the same birthday.
- With random $n$-bit strings, first collision after roughly $2^{n/2}$ draws.
- More generally, $2^{2t-n}$ collisions with $2^t$ draws

Security of CBC

- CBC leaks plaintext after $2^{n/2}$ blocks encrypted with the same key
- Security of mode can be lower than security of cipher
### Communication issues

<table>
<thead>
<tr>
<th>What cryptographers say</th>
<th>[Rogaway 2011]</th>
</tr>
</thead>
<tbody>
<tr>
<td>[Birthday] attacks can be a serious concern when employing a blockcipher of $n = 64$ bits, requiring relatively frequent rekeying to keep $\sigma \ll 2^{32}$</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>What standards say</th>
<th>[ISO SC27 SD12]</th>
</tr>
</thead>
<tbody>
<tr>
<td>The maximum amount of plaintext that can be encrypted before rekeying must take place is $2^{n/2}$ blocks, due to the birthday paradox. As long as the implementation of a specific block cipher do not exceed these limits, using the block cipher will be safe.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>What implementation do</th>
</tr>
</thead>
<tbody>
<tr>
<td>TLS libraries, web browsers</td>
</tr>
<tr>
<td>OpenVPN</td>
</tr>
</tbody>
</table>
## Communication issues

### What cryptographers say

[Birthday] attacks can be a serious concern when employing a blockcipher of $n = 64$ bits, requiring relatively frequent rekeying to keep $\sigma \ll 2^{32}$

### What standards say

The maximum amount of plaintext that can be encrypted before rekeying must take place is $2^{n/2}$ blocks, due to the birthday paradox. As long as the implementation of a specific block cipher do not exceed these limits, using the block cipher will be safe.

### What implementation do

- **TLS libraries, web browsers**: no rekeying
- **OpenVPN**: no rekeying (PSK mode) / rekey every hour (TLS mode)
Communication issues

What cryptographers say [Rogaway 2011]

[Birthday] attacks can be a serious concern when employing a blockcipher of \( n = 64 \) bits, requiring relatively frequent rekeying to keep \( \sigma \ll 2^{32} \).

What standards say [ISO SC27 SD12]

The maximum amount of plaintext that can be encrypted before rekeying must take place is \( 2^{n/2} \) blocks, due to the birthday paradox. As long as the implementation of a specific block cipher do not exceed these limits, using the block cipher will be safe.

What implementation do

TLS libraries, web browsers no rekeying
OpenVPN no rekeying (PSK mode) / rekey every hour (TLS mode)
Outline

Security and Cryptography

CBC Collision Attack

In Practice

MD5 Collisions

Breaking APOP

SLOTH Attack

Gaëtan Leurent (Inria) Security issues from bad crypto Journées pre-GDR sécurité 10 / 29
Block size is an important security parameter

- Block ciphers from the 90’s have a 64-bit block size
  - Blowfish, DES, 3DES
- Modern block ciphers have a 128-bit block size
  - AES, Twofish, CAMELLIA

- With $n = 64$, the bound is only 32 GB
- Around 1—2% of HTTPS connections use 3DES-CBC

<table>
<thead>
<tr>
<th></th>
<th>February 2016</th>
<th>October 2016</th>
<th>January 2017</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>support</td>
<td>use</td>
<td>support</td>
</tr>
<tr>
<td>Top 1k</td>
<td>93%</td>
<td>1.6%</td>
<td>84%</td>
</tr>
<tr>
<td>Top 1M</td>
<td>86%</td>
<td>1.3%</td>
<td>86%</td>
</tr>
</tbody>
</table>
Poorly configured websites

ebay.com
Poorly configured websites

match.com

Fixed in 2016
Poorly configured websites

match.com

https://discovery.cryptosense.com/analyze/208.83.241.15
**BEAST Attack Setting**

[Duong & Rizzo 2011]

- Attacker has access to the network (e.g., public WiFi)
  1. Attacker uses JS to generate traffic
     - Tricks victim to malicious site
     - JS makes *cross-origin* requests
  2. Attacker captures encrypted data

- Very powerful model
  Chosen plaintext

User >> Injests JS

Attacker

Captures encrypted traffic

Public WiFi
**BEAST collision attack**

- Assume user logged-in to secure website
- Javascript can generate HTTPS queries to secure website
- Each query includes an authentication token (cookie, password, ...)
  - HTTP is stateless

- Each collision reveals the xor of two plaintext blocks
- With some luck, xor of a known value and the secret

\[
\text{cookie} \oplus \text{header} = c_{i-1} \oplus c_{j-1}
\]

- Recover secret: \(\text{cookie} = \text{header} \oplus c_{i-1} \oplus c_{j-1}\)
### BEAST collision attack

The BEAST collision attack aims to exploit vulnerabilities in the CBC mode of operation by generating a plaintext and its corresponding ciphertext pair. The attack is illustrated as follows:

- **Plaintext**: GET `/index.html` HT TP/1.1 Cookie: `C=?? ???`
- **Ciphertexts**: $2^n/2-t/2$

The plaintext is chosen to be `GET `/index.html` HT TP/1.1 Cookie: `C=?? ???` and the corresponding ciphertexts are generated such that $2^n/2-t/2$.
**BEAST collision attack**

![Diagram showing plaintext and ciphertexts with a 2^t threshold]

- **Plaintext**
  - GET /index.html HT TP/1.1 Cookie:
  - GET 031 ED8 EEB 6CC B5A 440 067 154 AB5 CEE 015 70A 1ED 1B7
  - 2^n/2 - t/2

- **Ciphertexts**
  - 2^t
  - Gaëtan Leurent (Inria)

---

**Security and Cryptography**

- CBC Collision Attack
- In Practice
- MD5 Collisions
- APOP
- SLOTH
- Conclusion

**BEAST collision attack**

- Plaintext:
  - GET /index.html HT TP/1.1 Cookie:
  - 178 4E5 71A A39 68A 399 7D8 8F0 FEA 902 932 204 85A 969
  - E57 1AA 396 8A3 997 D88 F0F EA9 029 322 048 5A9 6E0 EA4
  - 1D6 645 EA2 050 FAE D74 A72 E5C 913 447 3B4 BAA 321 784
  - 7A5 322 700 DE3 BA8 7DD 998 040 A8D 9A2 05A EE5 330 9EC
  - 9BE 78D 350 AF5 327 311 F5B 252 77A C45 49E 2ED 20C 030

- Ciphertexts:
  - 289 597 BED 540 A60 7AF F96 511 AF2 41F 278 D25 400 4EB
  - 031 ED8 EEB 6CC B5A 440 067 154 AB5 CEE 015 70A 1ED 1B7
  - 38E 018 41A DEB 970 2D3 97A F0E 45C 94B 251 218 5FB 82A
  - 417 FF4 81D 00D 49D D9A 841 737 416 BA8 452 AC0 335 793
  - 21B B07 A20 4F4 C1D B07 2DF 410 340 6AB 0D2 96B CE9 4C9
  - 536 BDA A93 B85 351 831 763 FA0 E95 E5F 1EE 986 7D5 8C0
  - 5F5 935 574 21D EE0 1BF 338 6DB DDC F67 090 7F6 8EC A8D
**BEAST collision attack**

---

**Plaintext**

```plaintext
GET/unicode/html HT TP/1.1 Cookie:0C=?? ???
178 4E5 71A A39 68A 399 7D8 8F0 FEA 902 932 204 85A 969
E57 1AA 396 8A3 997 D88 F0F EA9 029 322 048 5A9 6E0 EA4
```

---

**Ciphertexts**

```plaintext
1D6 645 EA2 050 FAE D74 A72 E5C 913 447 3B4 BAA 321 784
7A5 322 700 DE3 BA8 7DD 998 040 A8D 9A2 05A EE5 330 9EC
9BE 78D 350 AF5 327 311 F5B 252 77A C45 49E 2ED 20C 030
289 597 BED 540 A60 7AF F96 511 AF2 41F 278 D25 400 4EB
031 ED8 EEB 6CC B5A 440 067 154 AB5 CEE 015 70A 1ED 1B7
38E 018 41A DEB 970 2D3 97A F0E 45C 94B 251 218 5FB 82A
417 FF4 81D 00D 49D D9A 841 737 416 BA8 452 AC0 335 793
21B B07 A20 4F4 C1D B07 2DF 410 340 6AB 0D2 96B CE9 4C9
536 BDA A93 B85 351 831 763 FA0 E95 E5F 1EE 986 7D5 8C0
5F5 935 574 21D EE0 1BF 338 6DB DDC F67 090 7F6 8EC A8D
```

---

Gaëtan Leurent (Inria)  
Security issues from bad crypto  
Journées pre-GDR sécurité
**BEAST collision attack**

plaintext

\[
\text{GET } /\text{i}n\text{de}x.h\text{tml } 1.1 \text{Cookie} : C = ?? ?? ??
\]

\[
\begin{array}{cccccccccccc}
178 & 4E5 & 71A & A39 & 68A & 399 & 7D8 & 8F0 & FEA & 902 & 932 & 204 & 85A & 969 \\
E57 & 1AA & 396 & 8A3 & 997 & D88 & F0F & EA9 & 029 & 322 & 048 & 5A9 & 6E0 & EA4 \\
1D6 & 645 & EA2 & 050 & FAE & D74 & A72 & E5C & 913 & 447 & 3B4 & BAA & 321 & 784 \\
\end{array}
\]

\[
\text{Ciphertexts}
\]

\[
\begin{array}{cccccccccccccccc}
7A5 & 322 & 700 & DE3 & BA8 & 7DD & 998 & 040 & A8D & 9A2 & 05A & EE5 & 330 & 9EC \\
9BE & 78D & 350 & AF5 & 327 & 311 & F5B & 252 & 77A & C45 & 49E & 2ED & 20C & 030 \\
289 & 597 & BED & 540 & A60 & 7AF & F96 & 511 & AF2 & 41F & 278 & D25 & 400 & 4EB \\
031 & ED8 & EEB & 6CC & B5A & 440 & 067 & 154 & AB5 & CEE & 015 & 70A & 1ED & 1B7 \\
38E & 018 & 41A & DEB & 970 & 2D3 & 97A & F0E & 45C & 94B & 251 & 218 & 5FB & 82A \\
417 & FF4 & 81D & 00D & 49D & D9A & 841 & 737 & 416 & BA8 & 452 & AC0 & 335 & 793 \\
21B & B07 & A20 & 4F4 & C1D & B07 & 2DF & 410 & 340 & 6AB & 0D2 & 96B & CE9 & 4C9 \\
536 & BDA & A93 & B85 & 351 & 831 & 763 & FA0 & E95 & E5F & 1EE & 986 & 7D5 & 8C0 \\
5F5 & 935 & 574 & 21D & EE0 & 1BF & 338 & 6DB & DDC & F67 & 090 & 7F6 & 8EC & A8D \\
\end{array}
\]
BEAST collision attack

Plaintext

GET /index.html HT TP/ 1.1 Cookie:\n
Ciphertexts

2^n/2−t/2

Gaëtan Leurent (Inria)

Security issues from bad crypto

Journées pre-GDR sécurité 14 / 29
BEAST collision attack

 Plaintext

\[
\text{GET} /i\text{n}\text{d}e\text{x}\text{.h}\text{tm}l \ HT TP/ 1.1 Coo\text{kie} :C = ?? ??
\]

178 4E5 71A A39 68A 399 7D8 8F0 FEA 902 932 204 85A 969
E57 1AA 396 8A3 997 D88 F0F EA9 029 322 048 5A9 6E0 EA4
1D6 645 EA2 050 FAE D74 A72 E5C 913 447 3B4 BAA 321 784
7A5 322 700 DE3 BA8 7DD 998 040 A8D 9A2 05A EE5 330 9EC
9BE 78D 350 AF5 327 311 F5B 252 77A C45 49E 2ED 20C 030

\[2^n/2 - t/2\]

Ciphertexts

\[
\begin{align*}
289 & \ 597 & \ BED & \ 540 & \ A60 & \ 7AF & \ F96 & \ 511 & \ AF2 & \ 41F & \ 278 & \ D25 & \ 400 & \ 4EB \\
031 & \ ED8 & \ EEB & \ 6CC & \ B5A & \ 440 & \ 067 & \ 154 & \ AB5 & \ CEE & \ 015 & \ 70A & \ 1ED & \ 1B7 \\
38E & \ 018 & \ 41A & \ DEB & \ 970 & \ 2D3 & \ 97A & \ F0E & \ 45C & \ 94B & \ 251 & \ 218 & \ 5FB & \ 82A \\
417 & \ FF4 & \ 81D & \ 00D & \ 49D & \ D9A & \ 841 & \ 737 & \ 416 & \ BA8 & \ 452 & \ AC0 & \ 335 & \ 793 \\
21B & \ B07 & \ A20 & \ 4F4 & \ C1D & \ B07 & \ 2DF & \ 410 & \ 340 & \ 6AB & \ 0D2 & \ 96B & \ CE9 & \ 4C9 \\
536 & \ BDA & \ A93 & \ B85 & \ 351 & \ 831 & \ 763 & \ FA0 & \ E95 & \ E5F & \ 1EE & \ 986 & \ 7D5 & \ 8C0 \\
5F5 & \ 935 & \ 574 & \ 21D & \ EE0 & \ 1BF & \ 338 & \ 6DB & \ DDC & \ F67 & \ 090 & \ 7F6 & \ 8EC & \ A8D \\
\end{align*}
\]
### BEAST collision attack

![BEAST collision attack diagram](image)

<table>
<thead>
<tr>
<th>Plaintext</th>
<th>Ciphertexts</th>
</tr>
</thead>
<tbody>
<tr>
<td>GET/indice.html</td>
<td>031 ED8 E6B 6CC B5A 440 067 154 AB5 CEE 015 70A 1ED 1B7</td>
</tr>
<tr>
<td></td>
<td>38E 018 41A DEB 970 2D3 97A FOE 45C 94B 251 218 5FB 82A</td>
</tr>
<tr>
<td></td>
<td>417 FF4 81D 00D 49D 9A 841 737 416 BA8 452 AC0 335 793</td>
</tr>
<tr>
<td></td>
<td>21B B07 A20 4F4 C1D B07 2DF 410 340 6AB 0D2 96B CE9 4C9</td>
</tr>
<tr>
<td></td>
<td>536 BDA A93 B85 351 831 763 FA0 E95 E5F 1EE 986 7D5 8C0</td>
</tr>
<tr>
<td></td>
<td>5F5 935 574 21D EE0 1BF 338 6DB DDC F67 090 7F6 8EC A8D</td>
</tr>
</tbody>
</table>

**Conclusion**

Gaëtan Leurent (Inria)  
Security issues from bad crypto  
Journées pre-GDR sécurité
BEAST collision attack

\[ 2^t \]

**Plaintext**

```
GET/index.html HT TP/1.1 Cookie:C==??
178 4E5 71A A39 68A 399 7D8 8F0 FEA 902 932 204 85A 969
E57 1AA 396 8A3 997 D88 F0F EA9 029 322 048 5A9 6E0 EA4
1D6 645 EA2 050 FAE D74 A72 E5C 913 447 3B4 BAA 321 784
7A5 322 700 DE3 BA8 7DD 998 040 A8D 9A2 05A EE5 330 9EC
9BE 78D 350 AF5 327 311 F5B 252 77A C45 49E 2ED 20C 030
289 597 BED 540 A60 7AF F96 511 AF2 41F 278 D25 400 4EB
```

\[ 2^{n/2-t/2} \]

**Ciphertexts**

```
031 ED8 EEB 6CC B5A 440 067 154 AB5 CEE 015 70A 1ED 1B7
```

Gaëtan Leurent (Inria)

Security issues from bad crypto

Journées pre-GDR sécurité
**BEAST collision attack**

\[ 2^n / 2 - t / 2 \]

**Plaintext**

GET \( / \) index \( . \) html \( \) HT TP/ 1.1 Cookie: \( \) ??

<table>
<thead>
<tr>
<th>178</th>
<th>4E5</th>
<th>71A</th>
<th>A39</th>
<th>68A</th>
<th>399</th>
<th>7D8</th>
<th>8F0</th>
<th>FEA</th>
<th>902</th>
<th>932</th>
<th>204</th>
<th>85A</th>
<th>969</th>
</tr>
</thead>
<tbody>
<tr>
<td>E57</td>
<td>1AA</td>
<td>396</td>
<td>8A3</td>
<td>997</td>
<td>D88</td>
<td>F0F</td>
<td>EA9</td>
<td>029</td>
<td>322</td>
<td>048</td>
<td>5A9</td>
<td>6E0</td>
<td>EA4</td>
</tr>
<tr>
<td>1D6</td>
<td>645</td>
<td>EA2</td>
<td>050</td>
<td>FAE</td>
<td>D74</td>
<td>A72</td>
<td>E5C</td>
<td>913</td>
<td>447</td>
<td>3B4</td>
<td>BAA</td>
<td>321</td>
<td>784</td>
</tr>
<tr>
<td>7A5</td>
<td>322</td>
<td>700</td>
<td>DE3</td>
<td>BA8</td>
<td>7DD</td>
<td>998</td>
<td>040</td>
<td>A8D</td>
<td>9A2</td>
<td>05A</td>
<td>EE5</td>
<td>330</td>
<td>9EC</td>
</tr>
<tr>
<td>9BE</td>
<td>78D</td>
<td>350</td>
<td>AF5</td>
<td>327</td>
<td>311</td>
<td>F5B</td>
<td>252</td>
<td>77A</td>
<td>C45</td>
<td>49E</td>
<td>2ED</td>
<td>20C</td>
<td>030</td>
</tr>
<tr>
<td>289</td>
<td>597</td>
<td>BED</td>
<td>540</td>
<td>A60</td>
<td>7AF</td>
<td>F96</td>
<td>511</td>
<td>AF2</td>
<td>41F</td>
<td>278</td>
<td>D25</td>
<td>400</td>
<td>4EB</td>
</tr>
<tr>
<td>031</td>
<td>ED8</td>
<td>EEB</td>
<td>6CC</td>
<td>B5A</td>
<td>440</td>
<td>067</td>
<td>154</td>
<td>AB5</td>
<td>CEE</td>
<td>015</td>
<td>70A</td>
<td>1ED</td>
<td>1B7</td>
</tr>
<tr>
<td>38E</td>
<td>018</td>
<td>41A</td>
<td>DEB</td>
<td>970</td>
<td>2D3</td>
<td>97A</td>
<td>F0E</td>
<td>45C</td>
<td>94B</td>
<td>251</td>
<td>218</td>
<td>5FB</td>
<td>82A</td>
</tr>
</tbody>
</table>

**Ciphertexts**

<table>
<thead>
<tr>
<th>417</th>
<th>FF4</th>
<th>81D</th>
<th>00D</th>
<th>49D</th>
<th>D9A</th>
<th>841</th>
<th>737</th>
<th>416</th>
<th>BA8</th>
<th>452</th>
<th>AC0</th>
<th>335</th>
<th>793</th>
</tr>
</thead>
<tbody>
<tr>
<td>21B</td>
<td>B07</td>
<td>A20</td>
<td>4F4</td>
<td>C1D</td>
<td>B07</td>
<td>2DF</td>
<td>410</td>
<td>340</td>
<td>6AB</td>
<td>0D2</td>
<td>96B</td>
<td>CE9</td>
<td>4C9</td>
</tr>
<tr>
<td>536</td>
<td>BDA</td>
<td>A93</td>
<td>B85</td>
<td>351</td>
<td>831</td>
<td>763</td>
<td>FA0</td>
<td>E95</td>
<td>E5F</td>
<td>1EE</td>
<td>986</td>
<td>7D5</td>
<td>8C0</td>
</tr>
<tr>
<td>5F5</td>
<td>935</td>
<td>574</td>
<td>21D</td>
<td>EE0</td>
<td>1BF</td>
<td>338</td>
<td>6DB</td>
<td>DDC</td>
<td>F67</td>
<td>090</td>
<td>7F6</td>
<td>8EC</td>
<td>A8D</td>
</tr>
</tbody>
</table>
## BEAST collision attack

### Plaintext

<table>
<thead>
<tr>
<th>GET</th>
<th>/</th>
<th>index</th>
<th>x.html</th>
<th>HT</th>
<th>TP/</th>
<th>1.1</th>
<th>Cookie</th>
<th>:</th>
<th>C</th>
<th>=??</th>
<th>???</th>
</tr>
</thead>
<tbody>
<tr>
<td>178</td>
<td>4E5</td>
<td>71A</td>
<td>A39</td>
<td>68A</td>
<td>399</td>
<td>7D8</td>
<td>8F0</td>
<td>FEA</td>
<td>902</td>
<td>932</td>
<td>204</td>
</tr>
<tr>
<td>E57</td>
<td>1AA</td>
<td>396</td>
<td>8A3</td>
<td>997</td>
<td>D88</td>
<td>F0F</td>
<td>EA9</td>
<td>029</td>
<td>322</td>
<td>048</td>
<td>5A9</td>
</tr>
<tr>
<td>1D6</td>
<td>645</td>
<td>EA2</td>
<td>050</td>
<td>FAE</td>
<td>D74</td>
<td>A72</td>
<td>E5C</td>
<td>913</td>
<td>447</td>
<td>3B4</td>
<td>BAA</td>
</tr>
<tr>
<td>7A5</td>
<td>322</td>
<td>700</td>
<td>DE3</td>
<td>BA8</td>
<td>7DD</td>
<td>998</td>
<td>040</td>
<td>A8D</td>
<td>9A2</td>
<td>05A</td>
<td>EE5</td>
</tr>
<tr>
<td>9BE</td>
<td>78D</td>
<td>350</td>
<td>AF5</td>
<td>327</td>
<td>311</td>
<td>F5B</td>
<td>252</td>
<td>77A</td>
<td>C45</td>
<td>49E</td>
<td>2ED</td>
</tr>
</tbody>
</table>

### Ciphertexts

<table>
<thead>
<tr>
<th>2^{n/2 - t/2}</th>
</tr>
</thead>
<tbody>
<tr>
<td>289</td>
</tr>
</tbody>
</table>

### Example

| 031 | ED8 | EEB | 6CC | B5A | 440 | 067 | 154 | AB5 | CEE | 015 | 70A | 1ED | 1B7 |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 38E | 018 | 41A | DEB | 970 | 2D3 | 97A | F0E | 45C | 94B | 251 | 218 | 5FB | 82A |
| 417 | FF4 | 81D | 00D | 49D | D9A | 841 | 737 | 416 | BA8 | 452 | AC0 | 335 | 793 |
**BEAST collision attack**

<table>
<thead>
<tr>
<th>Plaintext</th>
<th>Ciphertexts</th>
</tr>
</thead>
<tbody>
<tr>
<td>GET /index.html HT TP/1.1 Cookie : C = ?? ??</td>
<td></td>
</tr>
<tr>
<td>178 4E5 71A A39 68A 399 7D8 8F0 FEA 902 932 204 85A 969</td>
<td></td>
</tr>
<tr>
<td>E57 1AA 396 8A3 997 D88 F0F EA9 029 322 048 5A9 6E0 EA4</td>
<td></td>
</tr>
<tr>
<td>1D6 645 EA2 050 FAE D74 A72 E5C 913 447 3B4 BAA 321 784</td>
<td></td>
</tr>
<tr>
<td>7A5 322 700 DE3 BA8 7DD 998 040 A8D 9A2 05A EE5 330 9EC</td>
<td></td>
</tr>
<tr>
<td>9BE 78D 350 AF5 327 311 F5B 252 77A C45 49E 2ED 20C 030</td>
<td></td>
</tr>
<tr>
<td>289 597 BED 540 A60 7AF F96 511 AF2 41F 278 D25 400 4EB</td>
<td></td>
</tr>
<tr>
<td>031 ED8 EEB 6CC B5A 440 067 154 AB5 CEE 015 70A 1ED 1B7</td>
<td></td>
</tr>
<tr>
<td>38E 018 41A DEB 970 2D3 97A F0E 45C 94B 251 218 5FB 82A</td>
<td></td>
</tr>
<tr>
<td>417 FF4 81D 00D 49D D9A 841 737 416 BA8 452 AC0 335 793</td>
<td></td>
</tr>
<tr>
<td>21B B07 A20 4F4 C1D B07 2DF 410 340 6AB 0D2 96B CE9 4C9</td>
<td></td>
</tr>
</tbody>
</table>

\[2^n/2 - t/2\]
**BEAST collision attack**

<table>
<thead>
<tr>
<th>Plaintext</th>
<th>Ciphertexts</th>
</tr>
</thead>
<tbody>
<tr>
<td>GET /index .html HT TP/1.1 Cookie: C=?? ???</td>
<td></td>
</tr>
<tr>
<td>178 4E5 71A A39 68A 399 7D8 8F0 FEA 902 932 204 85A 969</td>
<td></td>
</tr>
<tr>
<td>E57 1AA 396 8A3 997 D88 F0F EA9 029 322 048 5A9 6E0 EA4</td>
<td></td>
</tr>
<tr>
<td>1D6 645 EA2 050 FAE D74 A72 E5C 913 447 3B4 BAA 321 784</td>
<td></td>
</tr>
<tr>
<td>7A5 322 700 DE3 BA8 7DD 998 040 A8D 9A2 05A EE5 330 9EC</td>
<td></td>
</tr>
<tr>
<td>9BE 78D 350 AF5 327 311 F5B 252 77A C45 49E 2ED 20C 030</td>
<td></td>
</tr>
<tr>
<td>289 597 BED 540 A60 7AF F96 511 AF2 41F 278 D25 400 4EB</td>
<td></td>
</tr>
<tr>
<td>031 ED8 EEB 6CC B5A 440 067 154 AB5 CEE 015 70A 1ED 1B7</td>
<td></td>
</tr>
<tr>
<td>38E 018 41A DEB 970 2D3 97A F0E 45C 94B 251 218 5FB 82A</td>
<td></td>
</tr>
<tr>
<td>417 FF4 81D 00D 49D D9A 841 737 416 BA8 452 AC0 335 793</td>
<td></td>
</tr>
<tr>
<td>21B B07 A20 4F4 C1D B07 2DF 410 340 6AB 0D2 96B CE9 4C9</td>
<td></td>
</tr>
<tr>
<td>536 BDA A93 B85 351 831 763 FA0 E95 E5F 1EE 986 7D5 8C0</td>
<td></td>
</tr>
</tbody>
</table>
**BEAST collision attack**

Plaintext: `GET /index.html HT TP/1.1 Cookie:`

Ciphertexts: `?? ?? ??`
Proof-of-concept Attack Demo

- Demo with **Firefox** (Linux), and **IIS 6.0** (Windows Server 2003)
  - Default configuration of IIS 6.0 does not support AES
  - Each HTTP request encrypted in TLS record, with fixed key

1. Generate traffic with malicious JavaScript
2. Capture on the network with `tcpdump`
3. Remove header, extract ciphertext at fixed position
4. Sort ciphertext (`stdxxl`), look for collisions

- **Expected time**: 38 hours for 785 GB (tradeoff q. size / # q.).
- **In practice**: 30.5 hours for 610 GB.

Another target

**OpenVPN** uses **Blowfish-CBC** by default
Comparison with RC4 attacks

Practical attacks against TLS with RC4

- With a different key each session
  - Using biases in the RC4 keystream
  - Plaintext recovery (220 first bytes) with $2^{28} - 2^{32}$ sessions
- With longer sessions
  - Using Fluhrer-McGrew biases (single or multiple sessions)
  - Cookie recovery with $2^{33} - 2^{34}$ requests
  - Latest improvement: $2^{30.2}$ requests [Vanhoef & Piessens, Usenix ’15]

Practical attack against TLS with 3DES

- Using a single long-lived session
- $2^{29.1}$ short query (512 bytes) 280 GB total
- Or $2^{27.6}$ longer queries (4 kB) 785 GB total
Disclosure

Sweet32 attack disclosed on August 24

- https://sweet32.info
- CVE-2016-2183, CVE-2016-6329

- OpenVPN 2.4 has cipher negotiation defaulting to AES
- Mozilla has implemented data limits in Firefox 51 (1M records)

Block size does matter

- Birthday attack against CBC with $2^{n/2}$ data
- Protocols from the 90’s still use 64-bit ciphers
- Attacks with $2^{32}$ data are practical
Outline

Security and Cryptography

CBC Collision Attack

In Practice

MD5 Collisions

Breaking APOP

SLOTH Attack
Hash Functions in Internet Protocols

- **Hash function**: public function \( \{0, 1\}^* \rightarrow \{0, 1\}^n \)
  - Maps arbitrary-length message to fixed-length hash

- **Security proofs assume collision-resistance.**

- **In practice, many protocols support weak functions**
  - TLS \( \leq 1.1 \) uses combinations of MD5 and SHA1
  - IKE, SSH use SHA1 (MD5 in some cases)
  - Hash-function negotiation for the signature added in TLS 1.2 (2008)
    - Introduces MD5 as an option...
Hash Functions in Internet Protocols

- Hash function: public function $\{0, 1\}^* \rightarrow \{0, 1\}^n$
  - Maps arbitrary-length message to fixed-length hash

- Security proofs assume collision-resistance.

- In practice, many protocols support weak functions
  - TLS $\leq 1.1$ uses combinations of MD5 and SHA1
  - IKE, SSH use SHA1 (MD5 in some cases)
  - Hash-function negotiation for the signature added in TLS 1.2 (2008)
    - Introduces MD5 as an option...
Hash function cryptanalysis

- Since 2005, attacks against widely used hash functions

\[
\begin{array}{|l|c|c|}
\hline
H & \text{Collision} & \text{CPC} \\
\hline
\text{Generic} & 2^{n/2} & 2^{n/2} \\
\text{MD5} & 2^{16} & 2^{39} \\
\text{SHA-1} & 2^{63} & 2^{77} \\
\text{MD5} \parallel \text{SHA-1} & 2^{67} & 2^{77} \\
\hline
\end{array}
\]

How bad is it?

- HMAC-MD5 is still mostly secure
- In most cases, the hash include fresh nonces
Collision attack

- Find $M_1 \neq M_2$ such that $H(M_1) = H(M_2)$
- Generic attack with complexity $2^{n/2}$ (expected security)
- Shortcut attacks
  - MD5: complexity $2^{16}$
  - SHA1: complexity $2^{63}$
- Arbitrary common prefix/suffix, random collision blocks
**Chosen-prefix collision attack**

- Given $P_1, P_2$, find $M_1 \neq M_2$ such that $H(P_1 \parallel M_1) = H(P_2 \parallel M_2)$
- Generic attack with complexity $2^{n/2}$ (expected security)
- Shortcut attacks
  - MD5: complexity $2^{39}$
  - SHA1: complexity $2^{77}$

$P_1$ $P_2$ $C_1$ $C'_1$ $C_2$ $C'_2$ $IV$ $S$ $S$

- Two different arbitrary prefixes

[Stevens & al. ’09]
[Stevens ’13]
Hash function cryptanalysis

- Since 2005, attacks against widely used hash functions

<table>
<thead>
<tr>
<th></th>
<th>Collision</th>
<th>CPC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Generic</td>
<td>$2^{n/2}$</td>
<td>$2^{n/2}$</td>
</tr>
<tr>
<td>MD5</td>
<td>$2^{16}$</td>
<td>$2^{39}$</td>
</tr>
<tr>
<td>SHA-1</td>
<td>$2^{63}$</td>
<td>$2^{77}$</td>
</tr>
<tr>
<td>MD5</td>
<td></td>
<td>SHA-1</td>
</tr>
</tbody>
</table>

How bad is it?

- HMAC-MD5 is still mostly secure
- In most cases, the hash include fresh nonces
Outline

Security and Cryptography

CBC Collision Attack

In Practice

MD5 Collisions

Breaking APOP

SLOTH Attack
APOP

Challenge-response authentication in POP3 mail protocol

- Man-in-the-middle can collect $\text{MD5}(x\|pw)$ for chosen $x$
- Can he recover the key?
APOP

- Challenge-response authentication in POP3 mail protocol
- Man-in-the-middle can collect $\text{MD5}(x\|pw)$ for chosen $x$
  - Can he recover the key?
Using collisions to recover the key

1. Guess the first password byte as $p^*$

2. Build a hash collision $(C_0, C_1)$ with $C_i = x_i || p^*$ (Full-block $C_0 \neq C_1$)
   
   \[
   C_1 = \underbrace{\ldots}_{\text{All identical}} p^* \\
   C_0 = \underbrace{\ldots}_{\text{All identical}} p^*
   \]

   \[
   x_1 = \underbrace{\ldots}_{\text{All identical}} \\
   x_0 = \underbrace{\ldots}_{\text{All identical}}
   \]

3. Send $x_1$ and $x_2$ as challenges and receive

   \[
   \text{MD5}(x_1 || pw) = \text{MD5}\left(\underbrace{\ldots}_{\text{All identical}} p_0 \right) \begin{array}{c} p_1 \ p_2 \ p_3 \ \ldots \end{array} \\
   \text{MD5}(x_0 || pw) = \text{MD5}\left(\underbrace{\ldots}_{\text{All identical}} p_0 \right) \begin{array}{c} p_1 \ p_2 \ p_3 \ \ldots \end{array}
   \]

4. If the guess was correct, collision after $p_0$

   ▶ With high probability $\text{MD5}(x_0 || p_0) \neq \text{MD5}(x_1 || p_0)$ if $p_0 \neq p^*$
   ▶ At most 256 attempts to recover $p_0$
   ▶ When $p_0$ known, attack $p_1$
In practice

Challenge format

- According to the RFC, the challenge is a message-id
  - Begins with ‘<’, end with ‘>’, single ‘@’ in the middle
  - Restricted set of characters (subset of ASCII)
- In practice, user agents enforced very few restrictions
- Since publication, strict checks limit attack [CVE-2007-1558]

Collision attack

- Need a strong collision attack
  - Control over the last bytes, with no message difference
- Variant of Wang’s attack recovers 3 characters [Leurent, FSE ’07]
- Attack based on dBB recovers 31 characters [Sasaki & al., RSA ’08]
Outline

Security and Cryptography

CBC Collision Attack

In Practice

MD5 Collisions

Breaking APOP

SLOTH Attack
Key exchange protocols

Diffie-Hellman key exchange

$$k = kdf(g^{xy} \mod p)$$
Key exchange protocols

Diffie-Hellman key exchange broken by Man in the Middle
Key exchange protocols

A

\[ m_1 = g^x \]

\[ m_2 = g^y \]

B

\[ k = \text{kdf}(g^{xy}) \]

\[ k = \text{kdf}(g^{xy}) \]

\[ \text{sign}(sk_A, m_1 \parallel m_2), \text{mac}(k, A) \]

\[ \text{sign}(sk_B, m_1 \parallel m_2), \text{mac}(k, B) \]

SIGMA protocol: authenticated DH (in practice) [Krawczyk '03]

- Add PKI: A knows \( sk_A, pk_b \), B knows \( sk_B, pk_A \)
- Sign transcript, prove knowledge of \( k \)
Key exchange protocols

SIGMA protocol: authenticated DH (in practice) [Krawczyk '03]

- Add info for parameters negotiation (flexible format)
- Signature uses a hash function (hash-and-sign)
Man-in-the-Middle attack against SIGMA'
Transcript collisions

Finds $x', y', info'_A, info'_B$ s.t.

$$h(g^x \| info_A \| g'^y \| info'_B) = h(g'^x \| info'_A \| g^y \| info_B)$$

1. If $g^y$ and $info_B$ are predictable, generic collision attack
   ▶ Complexity $2^{64}$ for MD5
Transcript collisions

Finds $x', y', \text{info}'_A, \text{info}'_B$ s.t.
\[ h(g^x \parallel \text{info}_A \parallel g^{y'} \parallel \text{info}'_B) = h(g^{x'} \parallel \text{info}'_A \parallel g^y \parallel \text{info}_B) \]

2. If no message boundaries in concatenation
   - Assume that garbage after info is ignored
   - Impersonate B with:

   $T_A = m_1 \parallel m_2 = g^x \parallel \text{info}_A \parallel g^{y'} \parallel \text{info}_M \parallel g^y \parallel \text{info}_B$

   $T_B = m_1' \parallel m_2 = g^x \parallel \text{info}_A \parallel g^{y'} \parallel \text{info}_M \parallel g^y \parallel \text{info}_B$

   - Forward signatures, compute A’s key with $g^{y'}$
Transcript collisions

Finds $x', y', info_A', info_B'$ s.t.
$$h(g^x || info_A || g'^y || info_B') = h(g'^x || info_A' || g^y || info_B)$$

3. If messages prefixed by message length
   - Assume that garbage after info is ignored
   - Use a chosen-prefix collision attack:
     $$\mathcal{T}_A = m_1 || m_2' = g^x || len_A || info_A || g'^y || len'_B || C_1 || g^y || len_B || info_B$$
     $$\mathcal{T}_B = m'_1 || m_2 = g'^x || len'_A || C_2 || g^y || len_B || info_B$$

   - Cost $\approx 2^{39}$ for MD5 (1 hour on 48 cores)
   - Cost $\approx 2^{77}$ for SHA1 or MD5 || SHA-1

[Stevens & al. '09]
[Stevens '13, Joux '04]
**TLS 1.2**

Client \( C \)

- \( \text{CH}(n_c, e_{x_C}) \)
- \( \text{SH}(n_s, e_{x_S}) \)
- \( \text{SC}(\text{cert}_s) \)
- \( \text{SKE}(\text{sign}(sk_S, \text{hash}(n_c \mid n_s \mid p \mid g \mid g^y))) \)
- \( \text{SCR}(dn) \)
- \( \text{SHD} \)
- \( \text{CC}(\text{cert}_c) \)
- \( \text{CKE}(g^x) \)
- \( \text{CCV}(\text{sign}(sk_C, \text{hash}(log_1))) \)

Server \( S \)

- \( \text{kdf}(g^{xy}, n_c \mid n_s) \)
- \( \text{NPN}(e_{x_n})^{k_1} \)
- \( \text{CFIN}(%mac_{96}(ms, \text{hash}(log_2)))^{k_1} \)
- \( \text{SFIN}(%mac_{96}(ms, \text{hash}(log_3)))^{k_2} \)

\[(ms, k_1, k_2) = \text{kdf}(g^{xy}, n_c \mid n_s)\]
Server directly signs nonce and DH parameters (not transcript)

- Cannot use transcript collisions for server impersonation
- On the other hand, this allows LogJam...

Client sends $g^x$ and signature together

- No flexible message after sending $g^x$
- SIGMA attack not applicable as is
Breaking client authentication in TLS 1.2

- Assume client connects to $M$, authenticates with certificate also used for $S$.
- We make the client DH share **predictable in a bogus group**
  - With $p = g^2 - g$ (not prime), $\forall x, g^x \equiv g \mod p$
- We can stuff data in
  - ClientHello extensions ($C \rightarrow S$)
  - CertificateRequest list of accepted CA ($S \rightarrow C$)

\[
\mathcal{T}_C = \text{CH} || \text{SH}' || \text{SC'} || \text{SKE'} || \text{SCR}(C_1, \text{SH} || \text{SC} || \text{SKE} || \text{SCR})
\]
\[
\mathcal{T}_S = \text{CH}(n_C, C_2) || \text{SH} || \text{SC} || \text{SKE} || \text{SCR}
\]
- Forward the client signature,
  Finish connection with known DH keys
Breaking client authentication in TLS 1.2

Computes $ex'_c, dn'$ s.t. $hash(log^c_i) = hash(log^s_i)$ by finding a chosen-prefix collision $(C_1, C_2)$ s.t.:

$hash(CH | SH' | SC' | SKE' | SCR'(C_1 | -)) = hash(CH'(n_c, C_2))$

Authenticated Connection: $C \rightarrow S$
**SLOTH Attack**

SLOTH : Security Losses from Obsolete and Truncated Transcript Hashes


- We show a class of transcript collision attack
  - Man-in-the-middle can tamper with the key exchange messages
  - If messages collide, signature still valid

- MD5 is still in standards
- Collision attacks do break key-exchange
  - Almost practical client impersonation for TLS 1.2 with MD5
- Also applications to SSH and IKE

- TLS libraries removed support for MD5 signatures
Conclusion

Sweet32 : On the Practical (In-)Security of 64-bit Block Ciphers
Bhargavan, G. L.  
[ACM CCS ’16]

Message Freedom in MD4 and MD5 Collisions : Application to APOP
G. L.  
[FSE ’07]

Transcript Collision Attacks : Breaking Authentication in TLS, IKE, and SSH
Bhargavan, G. L.  
[NDSS ’16]

Practical impact of cryptanalysis

- When proofs don’t apply, attacks become possible
  - It can be hard to evaluate the practical impact of attacks
  - Better safe than sorry?

- Practical demonstration of attacks help convince users
CBC vs CTR mode

CBC mode

- Security proof up to the birthday bound
- Collisions reveals xor of two plaintext blocks

CTR mode

- Security proof up to the birthday bound
- Distinguishing attack: Key stream doesn't collide
Support slides

**TLS cipher use in Firefox (telemetry)**

- **2014**
- **2015**
- **2016**

FF 36 disables RC4

- 3DES
- RC4
- AES

Gaëtan Leurent (Inria) Security issues from bad crypto Journées pre-GDR sécurité
**TLS cipher use in Firefox (telemetry)**

- **2014**: 10% 3DES, 0% RC4, 0% AES
- **2015**: 6% 3DES, 0% RC4, 94% AES
- **2016**: 4% 3DES, 0% RC4, 96% AES

FF 36 disables RC4