Security issues from bad crypto

Gaëtan Leurent

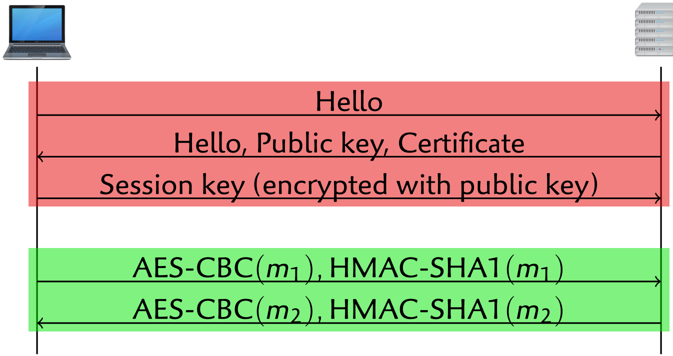
Joint work with:
Karthikeyan Bhargavan

Inria

Journées pre-GDR sécurité

Secure channel (TLS)

- Crypto provides secure communication against an adversary



- 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

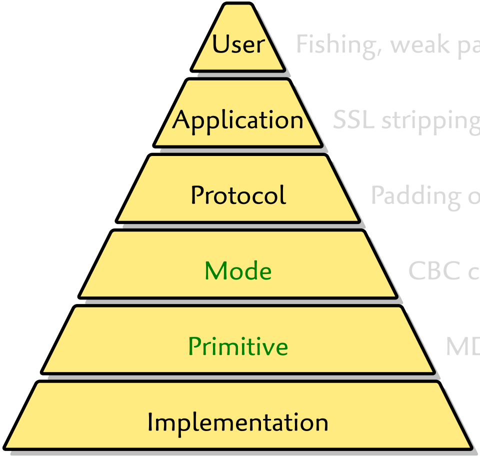
- Security of the protocol
  - Security proofs assuming security of cryptographic operations
- Security of the modes (HMAC, CBC, ...)
  - Security proofs (assuming security of the primitive)
- Security of the primitives (AES, SHA-1, RSA, ...)
  - Studied with cryptanalysis

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 mostlty know how to build good crypto...



Fishing, weak passwords, password reuse, ...

SSL stripping, bad cert checks, encryption only, ...

Padding oracle, predictible IV (BEAST)

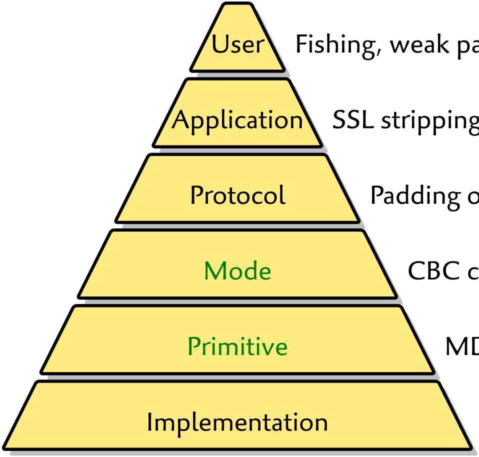
CBC collisions (Sweet32), CTR malleability (To

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 mostlty know how to build good crypto...



Fishing, weak passwords, password reuse, ...

SSL stripping, bad cert checks, encryption only, ...

Padding oracle, predictible IV (BEAST)

CBC collisions (Sweet32), CTR malleability (To

MD5 collisions, RC4 bias

Side channel, buffer overflow, bugs (SM

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

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

For users

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

Attacks only get better

For cryptographers

- Attack primitive
- If broken, stop using it
  - Proof hypothesis broken

For users

- Does it break real protocols ?
- Migration is expensive

Cryptanalysis in theory and in practice

Cryptanalysis of MD5

1993	Compression function attack		
2005	Collision attack	→ *2007*	Exploitable in APOP
2007	Free-start collision attack	→ *2009*	Exploitable for rogue CA
		↪ *2013*	Exploited by Flame

Cryptanalysis of RC4

2000	Biases in RC4 keystream	→ *2013*	Exploitable in TLS
2001	Related-key attack on RC4	→ *2002*	Exploitable in WEP

This talk

- Leverage weakness of crypto algorithms to break protocols

Cryptanalysis in theory and in practice

Cryptanalysis of MD5

1993	Compression function attack		
2005	Collision attack	→ 2007	Exploitable in APOP
2007	Free-start collision attack	→ 2009	Exploitable for rogue CA
		↪ 2013	Exploited by Flame

Cryptanalysis of RC4

2000	Biases in RC4 keystream	→ 2013	Exploitable in TLS
2001	Related-key attack on RC4	→ 2002	Exploitable in WEP

This talk

- Leverage weakness of crypto algorithms to break protocols

Outline

Outline

Security and Cryptography

CBC Collision Attack

In Practice

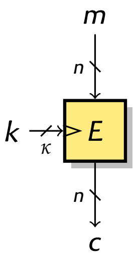
MD5 Collisions

Breaking APOP

SLOTH Attack

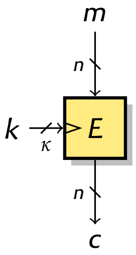
Block ciphers and Modes of operation

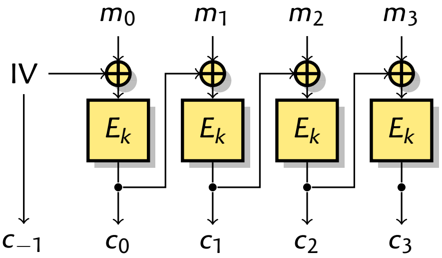
- A block cipher is a family of permutations
- It is used with a mode of operation : CBC, CTR, GCM, ...
  - To deal with variable-length messages
  - To include randomness
  - Important example : CBC



Block ciphers and Modes of operation

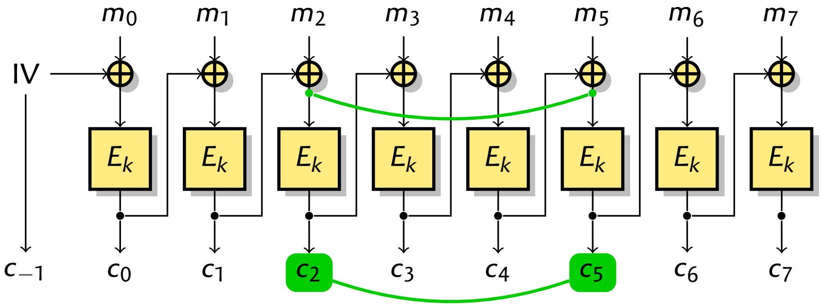
- A block cipher is a family of permutations
- It is used with a mode of operation : CBC, CTR, GCM, ...
  - To deal with variable-length messages
  - To include randomness
  - Important example : CBC





CBC collisions

- Well known collision attack against CBC

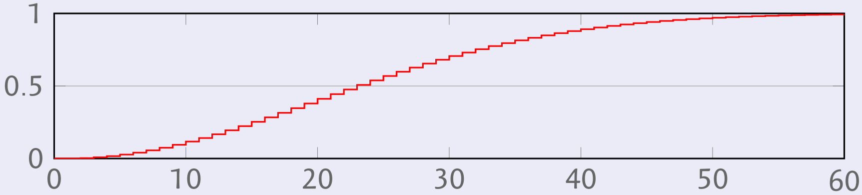


- If $c_i = c_j$, then $c_{i-1} \oplus m_i = c_{j-1} \oplus m_j$
- Ciphertext collision reveals the xor of two plaintext blocks

Birthday paradox

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

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)

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)

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

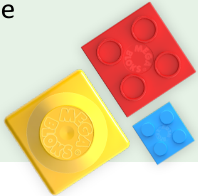
Breaking APOP

SLOTH Attack

Block size

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

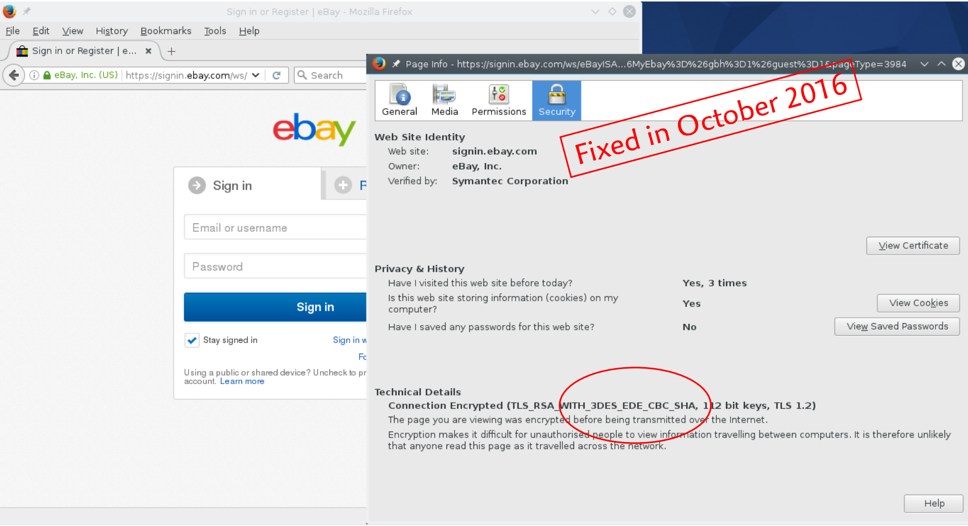


- With $n = 64$, the bound is only 32 GB
- Around 1—2% of HTTPS connections use 3DES-CBC

	February 2016		October 2016		January 2017	
3DES	support	use	support	use	support	use
Top 1k	93%	1.6%	84%	1.5%	75%	1.1%
Top 1M	86%	1.3%	86%	1.0%		

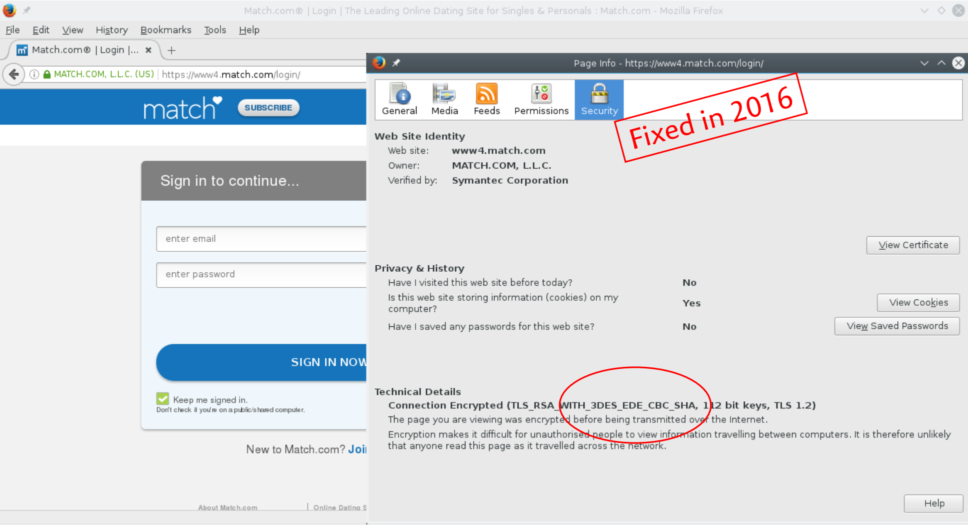
Poorly configured websites

ebay.com



Poorly configured websites

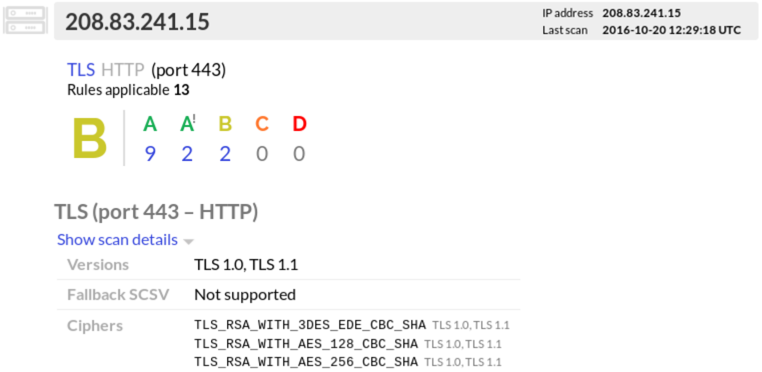
match.com



Poorly configured websites

match.com

`https://discovery.cryptosense.com/analyze/208.83.241.15`



208.83.241.15

IP address 208.83.241.15
Last scan 2016-10-20 12:29:18 UTC

TLS HTTP (port 443)
Rules applicable **13**

B

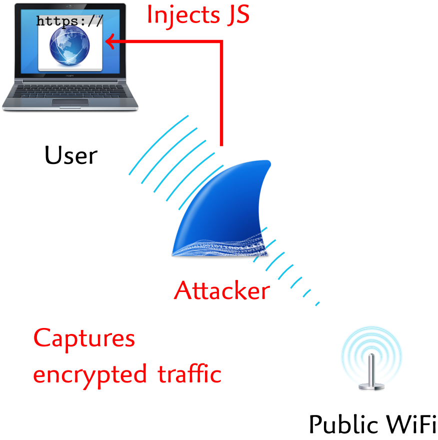
A	A!	B	C	D
9	2	2	0	0

TLS (port 443 – HTTP)

Show scan details

Versions	TLS 1.0, TLS 1.1
Fallback SCSV	Not supported
Ciphers	TLS_RSA_WITH_3DES_EDE_CBC_SHA TLS 1.0, TLS 1.1 TLS_RSA_WITH_AES_128_CBC_SHA TLS 1.0, TLS 1.1 TLS_RSA_WITH_AES_256_CBC_SHA TLS 1.0, TLS 1.1

BEAST Attack Setting [Duong & Rizzo 2011]



- Attacker has access to the network (*eg.* 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

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

$$\underbrace{\texttt{cookie}}_{\text{unknown}} \oplus \underbrace{\texttt{header}}_{\text{known}} = \underbrace{c_{i-1} \oplus c_{j-1}}_{\text{known}}$$

  - Recover secret : $\texttt{cookie} = \texttt{header} \oplus c_{i-1} \oplus c_{j-1}$

BEAST collision attack

2^t

Plaintext		GET	␣/i	nde	x.h	tml	␣HT	TP/	1.1	Coo	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
$2^{n/2-t/2}$	289	597	BED	540	A60	7AF	F96	511	AF2	41F	278	D25	400	4EB
Ciphertexts	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

2^t

Plaintext		GET	␣/i	nde	x.h	tml	␣HT	TP/	1.1	Coo	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
$2^{n/2-t/2}$	289	597	BED	540	A60	7AF	F96	511	AF2	41F	278	D25	400	4EB
Ciphertexts	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

2^t

Plaintext

	GET	␣/i	nde	x.h	tml	␣HT	TP/	1.1	Coo	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
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

$2^{n/2-t/2}$ Ciphertexts

BEAST collision attack

2^t

Plaintext		GET	␣/i	nde	x.h	tml	␣HT	TP/	1.1	Coo	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
$2^{n/2-t/2}$	289	597	BED	540	A60	7AF	F96	511	AF2	41F	278	D25	400	4EB
Ciphertexts	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

2^t

Plaintext	GET	␣/i	nde	x.h	tml	␣HT	TP/	1.1	Coo	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
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

$2^{n/2-t/2}$ Ciphertexts

BEAST collision attack

2^t

Plaintext	GET	␣/i	nde	x.h	tml	␣HT	TP/	1.1	Coo	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
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

$2^{n/2-t/2}$ Ciphertexts

BEAST collision attack

2^t

Plaintext	GET	␣/i	nde	x.h	tml	␣HT	TP/	1.1	Coo	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
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

$2^{n/2-t/2}$ Ciphertexts

BEAST collision attack

2^t

Plaintext		GET	␣/i	nde	x.h	tml	␣HT	TP/	1.1	Coo	kie	:␣C	=??	???
$2^{n/2-t/2}$ *Ciphertexts*	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
	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

2^t

Plaintext

$2^{n/2-t/2}$ Ciphertexts

	GET	␣/i	nde	x.h	tml	␣HT	TP/	1.1	Coo	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
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

2^t

Plaintext	GET	␣/i	nde	x.h	tml	␣HT	TP/	1.1	Coo	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
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

$2^{n/2-t/2}$ Ciphertexts

BEAST collision attack

2^t

Plaintext

	GET	␣/i	nde	x.h	tml	␣HT	TP/	1.1	Coo	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
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

$2^{n/2-t/2}$ Ciphertexts

BEAST collision attack

2^t

$2^{n/2-t/2}$ Ciphertexts

Plaintext		GET	␣/i	nde	x.h	tml	␣HT	TP/	1.1	Coo	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
	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

2^t

Plaintext	GET	␣/i	nde	x.h	tml	␣HT	TP/	1.1	Coo	kie	:␣C	=??	???
$2^{n/2-t/2}$ *Ciphertexts*	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
	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

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 *[AFBPPS, Usenix '13]*

- 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

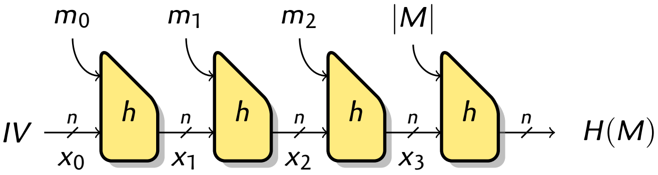
- OpenVPN 2.4 has cipher negotiation defaulting to AES
- Mozilla has implemented data limits in Firefox 51 (1M records)

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

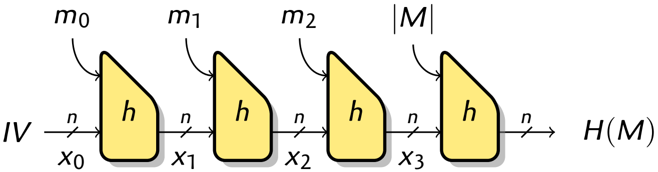
Outline

Hash Functions in Internet Protocols



- Hash function : public function $\{0,1\}^* \to \{0,1\}^n$
  - 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 $\{0,1\}^* \to \{0,1\}^n$
  - 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

H	Collision	CPC
Generic	$2^{n/2}$	$2^{n/2}$
MD5	2^{16}	2^{39}
SHA-1	2^{63}	2^{77}
MD5 $\|$SHA-1	2^{67}	2^{77}

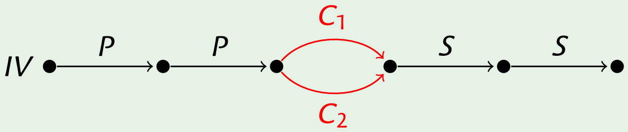
How bad is it ?

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

Hash function cryptanalysis

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} [Wang & *al.* '05, Stevens & *al.* '09]
  - SHA1 : complexity 2^{63} [Wang & *al.* '05, Stevens & *al.* '17]

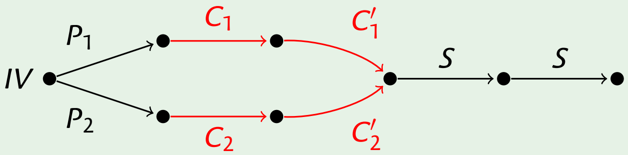


- Arbitrary common prefix/suffix, random collision blocks

Hash function cryptanalysis

Chosen-prefix collision attack

- Given P_1, P_2, find $M_1 \neq M_2$ such that $H(P_1 \| M_1) = H(P_2 \| M_2)$
- Generic attack with complexity $2^{n/2}$ (expected security)
- Shortcut attacks
  - MD5 : complexity 2^{39} [Stevens & *al.* '09]
  - SHA1 : complexity 2^{77} [Stevens '13]



- Two different arbitrary prefixes

Hash function cryptanalysis

- Since 2005, attacks against widely used hash functions

H	Collision	CPC
Generic	$2^{n/2}$	$2^{n/2}$
MD5	2^{16}	2^{39}
SHA-1	2^{63}	2^{77}
MD5 $\|$ SHA-1	2^{67}	2^{77}

How bad is it ?

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

Outline

APOP

Alice

password *pw*

$\xrightarrow{\quad id \quad}$

$\boxed{x \leftarrow \$}$

$\xleftarrow{\quad x \quad}$

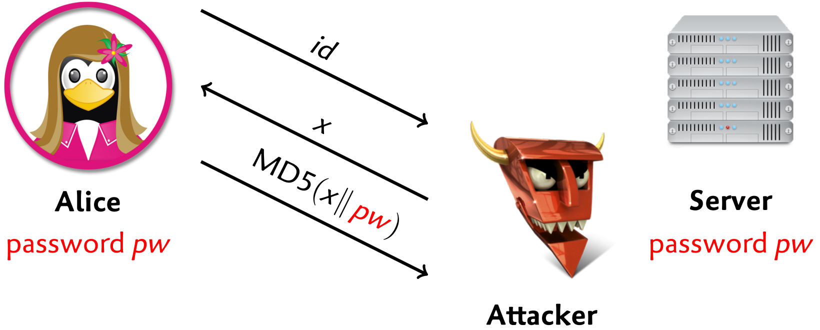
$\xrightarrow{\quad \mathrm{MD5}(x \| pw) \quad}$

Server

password *pw*

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

APOP



- Challenge-response authentication in POP3 mail protocol
- Man-in-the-middle can collect $\mathrm{MD5}(x \| pw)$ 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 $C_0 \neq C_1$)

   $C_1 =$ | ‽‽‽···‽‽‽ | p^* | $x_1 =$ ‽‽‽···‽‽‽

   $C_0 =$ | ⸘⸘⸘···⸘⸘⸘ | p^* | $x_0 =$ ⸘⸘⸘···⸘⸘⸘

3. Send x_1 and x_2 as challenges and receive

   $\mathrm{MD5}(x_1 \| pw) = \mathrm{MD5}($ | ‽‽‽···‽‽‽ | p_0 | $p_1 p_2 p_3 \cdots$ | $)$

   $\mathrm{MD5}(x_0 \| pw) = \mathrm{MD5}($ | ⸘⸘⸘···⸘⸘⸘ | p_0 | $p_1 p_2 p_3 \cdots$ | $)$

4. If the guess was correct, collision after p_0

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

In practice

Challenge format

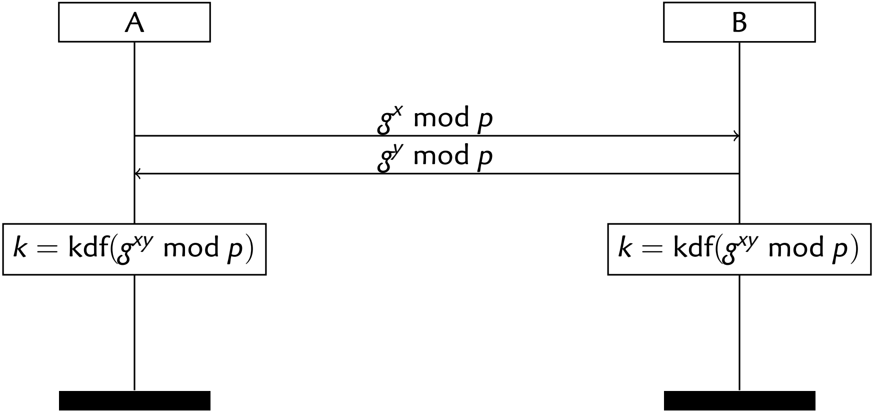
- According to the RFC, the challenge is a message-id
  - Begins with ‘<’, end with ‘>’, single ‘@’ in the middle
  - Restricted set of characters (subset of ASCII)
- In practice, user agents enforced very few restrictions
- Since publication, strict checks limit attack [CVE-2007-1558]

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]

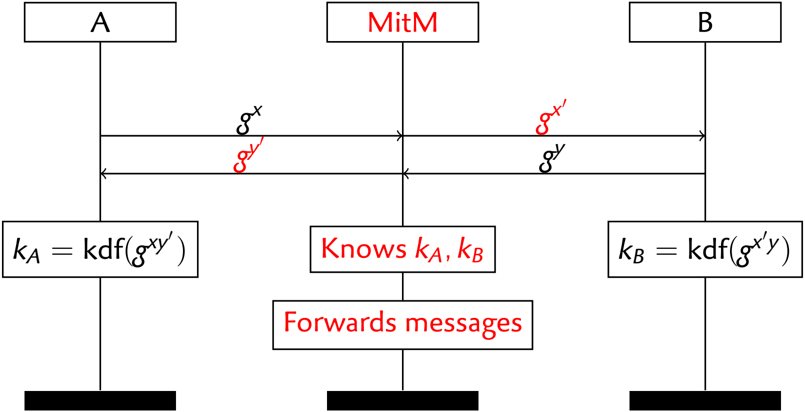
Outline

Key exchange protocols



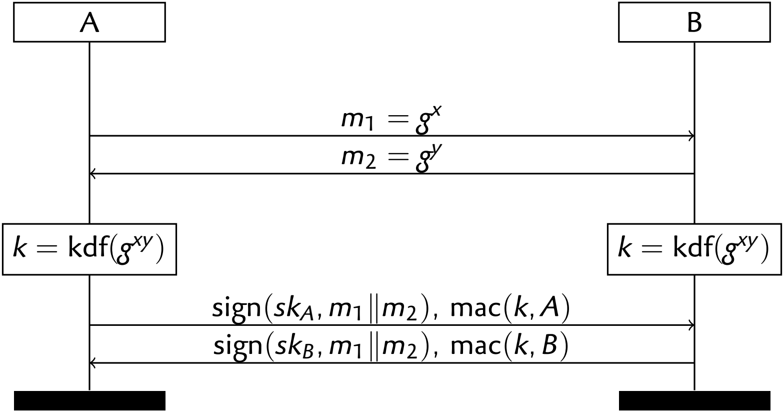
Diffie-Hellman key exchange

Key exchange protocols



Diffie-Hellman key exchange broken by Man in the Middle

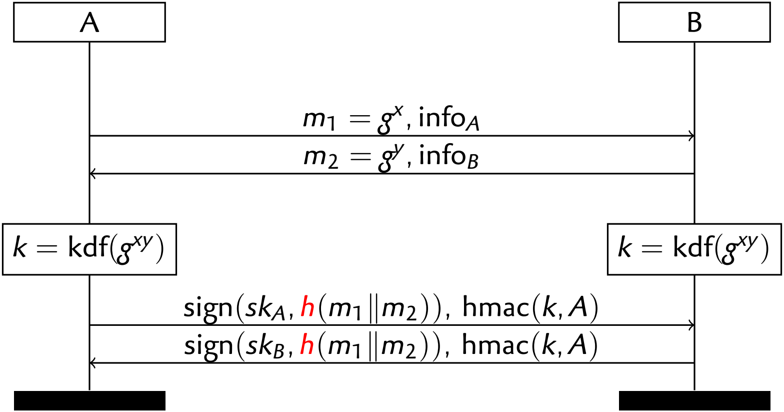
Key exchange protocols



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

- Add PKI : A known sk_A, pk_b, B knows sk_B, pk_A
- Sign transcript, prove knowledge of k

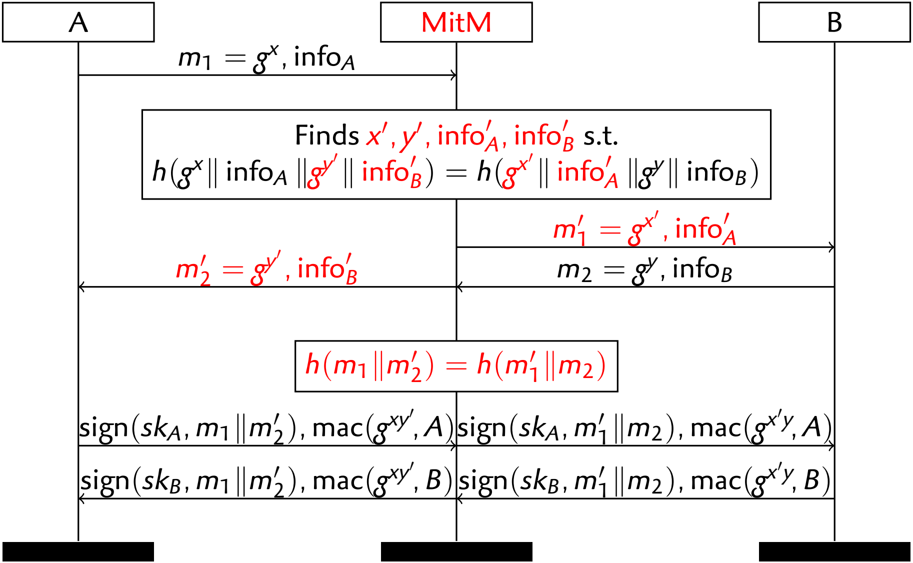
Key exchange protocols



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

$$\boxed{\begin{array}{c}\text{Finds } x', y', \text{info}'_A, \text{info}'_B \text{ s.t.}\\ h(g^x \,\|\, \text{info}_A \,\|\, g^{y'} \,\|\, \text{info}'_B) = h(g^{x'} \,\|\, \text{info}'_A \,\|\, g^y \,\|\, \text{info}_B)\end{array}}$$

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

Transcript collisions

$$\boxed{\begin{array}{c}\text{Finds } x', y', \text{info}'_A, \text{info}'_B \text{ s.t.}\\ h(g^x \,\|\, \text{info}_A \,\|\, g^{y'} \,\|\, \text{info}'_B) = h(g^{x'} \,\|\, \text{info}'_A \,\|\, g^y \,\|\, \text{info}_B)\end{array}}$$

2 If no message boundaries in concatenation

- Assume that garbage after info is ignored
- Impersonate B with :

$$\mathcal{T}_A = m_1 \| m'_2 = g^x \,\|\, \text{info}_A \,\|\, g^{y'} \,\|\, \text{info}_M \,\|\, \underbrace{g^y \,\|\, \text{info}_B}_{\text{info}'_B}$$

$$\mathcal{T}_B = m'_1 \| m_2 = g^x \,\|\, \underbrace{\text{info}_A \,\|\, g^{y'} \,\|\, \text{info}_M}_{\text{info}'_A} \,\|\, g^y \,\|\, \text{info}_B$$

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

Transcript collisions

$$\boxed{\begin{array}{c}\text{Finds } x', y', \text{info}'_A, \text{info}'_B \text{ s.t.}\\ h(g^x \,\|\, \text{info}_A \,\|\, g^{y'} \,\|\, \text{info}'_B) = h(g^{x'} \,\|\, \text{info}'_A \,\|\, g^y \,\|\, \text{info}_B)\end{array}}$$

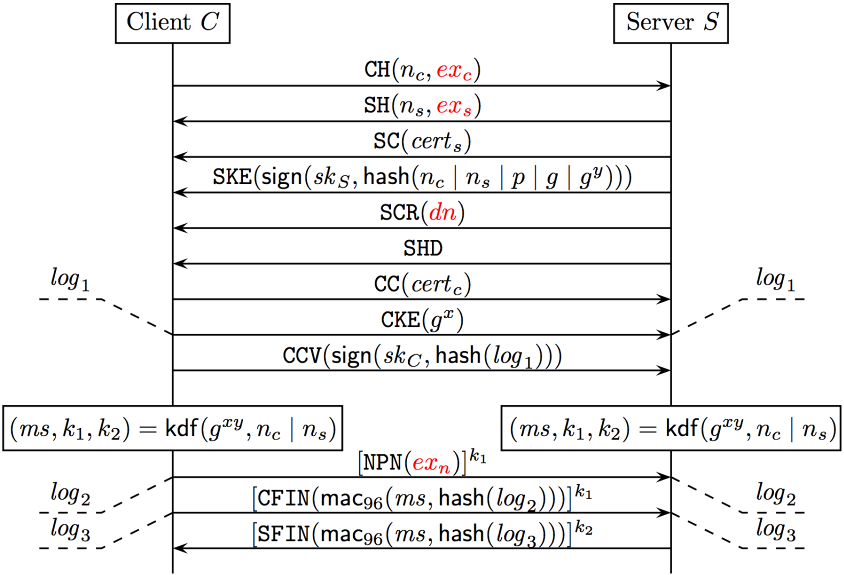
3. If messages prefixed by message length
   - Assume that garbage after info is ignored
   - Use a chosen-prefix collision attack :

$$\mathcal{T}_A = m_1 \| m'_2 = g^x \,\|\, \text{len}_A \,\|\, \text{info}_A \,\|\, g^{y'} \,\|\, \text{len}'_B \,\|\, \underbrace{C_1 \,\|\, g^y \,\|\, \text{len}_B \,\|\, \text{info}_B}_{\text{info}'_B}$$

$$\mathcal{T}_B = m'_1 \| m_2 = g^{x'} \,\|\, \text{len}'_A \,\|\, \underbrace{\qquad\qquad C_2}_{\text{info}'_A} \,\|\, g^y \,\|\, \text{len}_B \,\|\, \text{info}_B$$

- Cost $\approx 2^{39}$ for MD5 (1 hour on 48 cores) [Stevens & *al.* '09]
- Cost $\approx 2^{77}$ for SHA1 or MD5 $\|$SHA-1 [Stevens '13, Joux '04]

TLS 1.2


Client C
Server S
$\mathtt{CH}(n_c, ex_c)$
$\mathtt{SH}(n_s, ex_s)$
$\mathtt{SC}(cert_s)$
$\mathtt{SKE}(\mathsf{sign}(sk_S, \mathsf{hash}(n_c \mid n_s \mid p \mid g \mid g^y)))$
$\mathtt{SCR}(dn)$
$\mathtt{SHD}$
log_1
$\mathtt{CC}(cert_c)$
$\mathtt{CKE}(g^x)$
$\mathtt{CCV}(\mathsf{sign}(sk_C, \mathsf{hash}(log_1)))$
log_1
$(ms, k_1, k_2) = \mathsf{kdf}(g^{xy}, n_c \mid n_s)$
$(ms, k_1, k_2) = \mathsf{kdf}(g^{xy}, n_c \mid n_s)$
$[\mathtt{NPN}(ex_n)]^{k_1}$
log_2
$[\mathtt{CFIN}(\mathsf{mac}_{96}(ms, \mathsf{hash}(log_2)))]^{k_1}$
log_2
log_3
$[\mathtt{SFIN}(\mathsf{mac}_{96}(ms, \mathsf{hash}(log_3)))]^{k_2}$
log_3

TLS 1.2

- 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

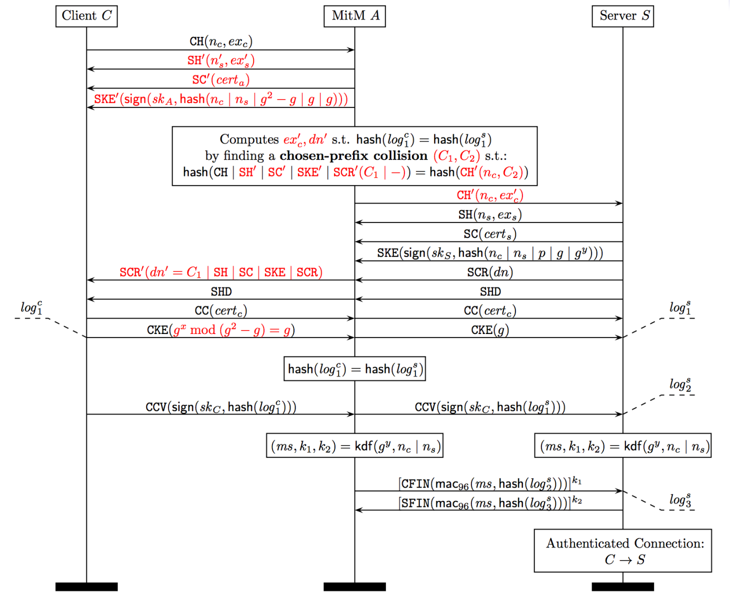
Breaking client authentication in TLS 1.2

- 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), $\forall x,\ g^x \equiv g \bmod p$
- We can stuff data in
  - `ClientHello` extensions ($C \to S$)
  - `CertificateRequest` list of accepted CA ($S \to C$)

$$\mathcal{T}_C = \mathtt{CH} \| \mathtt{SH}' \| \mathtt{SC}' \| \mathtt{SKE}' \| \mathtt{SCR}(C_1, \mathtt{SH} \| \mathtt{SC} \| \mathtt{SKE} \| \mathtt{SCR})$$
$$\mathcal{T}_S = \mathtt{CH}(n_C, \quad C_2) \| \mathtt{SH} \| \mathtt{SC} \| \mathtt{SKE} \| \mathtt{SCR}$$

- Forward the client signature,
  Finish connection with known DH keys

Breaking client authentication in TLS 1.2



SLOTH Attack

SLOTH : Security Losses from Obsolete and Truncated Transcript Hashes

https://www.mitls.org/pages/attacks/SLOTH [CVE-2015-7575]

- 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

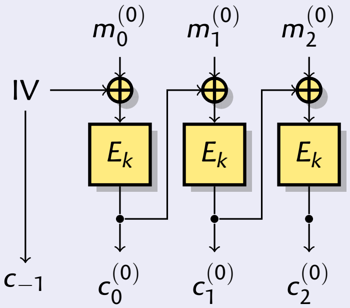
- Sweet32 : On the Practical (In-)Security of 64-bit Block Ciphers
  Bhargavan, G. L. [ACM CCS '16]
- Message Freedom in MD4 and MD5 Collisions : Application to APOP
  G. L. [FSE '07]
- Transcript Collision Attacks : Breaking Authentication in TLS, IKE, and SSH
  Bhargavan, G. L. [NDSS '16]

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

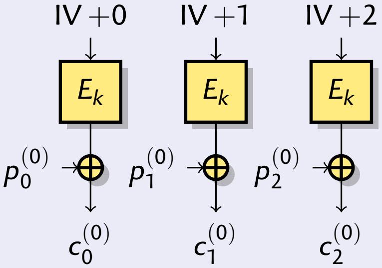
CBC vs CTR mode

CBC mode



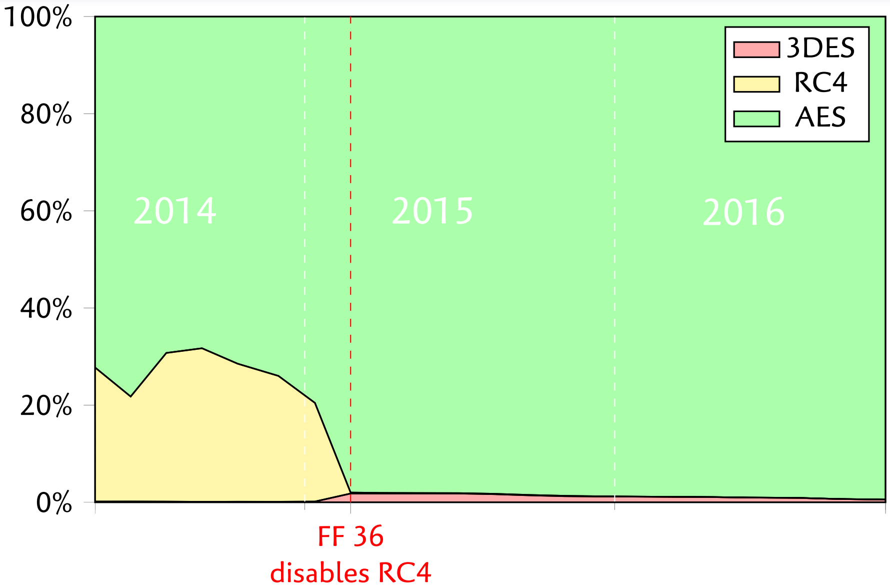
- Security proof up to the birthday bound
- Collisions reveals
  xor of two plaintext blocks

CTR mode



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

TLS cipher use in Firefox (telemetry)



TLS cipher use in Firefox (telemetry)