Security Issues with Small Block Sizes

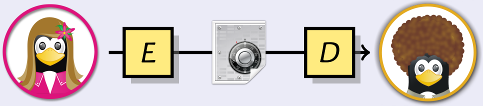
Gaëtan Leurent
Joined work with Karthikeyan Bhargavan, Ferdinand Sibleyras

Inria, France

Lightweight Crypto Day 2018

Confidentiality and authenticity

Confidentiality



c

$c = E_k(m)$ $\qquad$ $m = D_k(c)$

- Keeping the message secret
  - Adversary learns nothing about m
- Encryption
  - Block ciphers
  - Stream ciphers

Authenticity

M, t

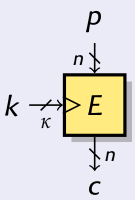
$t = \mathrm{MAC}_k(m)$ $\qquad$ $t \stackrel{?}{=} \mathrm{MAC}_k(m)$

- Make sure the message is authentic
  - Adversary cannot forge t
- Message Authentication Codes
  - From block ciphers
  - From hash functions
  - Dedicated, ...

Symmetric key primitives

Block cipher

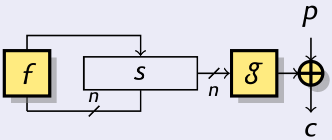
- Encrypt small block of message
- → PRP



- Iterate round function

Stream cipher

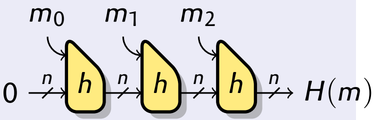
- Generate pseudo-random keystream from key
- → PRG



- Initialize state from key
- Update state, Generate keystream

Hash function

- Compress message to small digest
- → Random oracle



- Divide msg into blocks
- Iter. compression func.

- *Eg* DES, Blowfish, AES
- *Eg* RC4, Salsa20, Grain
- *Eg* MD5, SHA1/2/3

Going lightweight

Lightweight crypto (today)

Symmetric-key cryptography targeting low-end devices

- Low gate-count
- Low power / energy
- Low latency
- Optimized for micro-controllers
- Optimized for side-channel protection
- ...

- How to reduce the implementation cost?
  - Optimize for a specific constraint/platform
  - Reduced security margins
  - Reduced block size (often 64 bits)
- We have many candidates for lightweight block ciphers:
  - HIGHT (ISO std.)
  - CLEFIA (ISO std.)
  - PRESENT (ISO std.)
  - KASUMI (3GPP std.)
  - 3DES (former std.)
  - Noekeon
  - KATAN & KTANTAN
  - LBlock
  - PRINCE
  - Simon & Speck (NSA)
  - Robin & Fantomas
  - Skinny, ...

Security evaluation

Security goal

- As good as ideal primitive with the same parameters
  - Best attacks should be generic attacks
- Cryptanalysis to evaluate the concrete security
  - Broken: DES, GOST, KeeLoq, A5/1, RC4, MD5, SHA1, ...

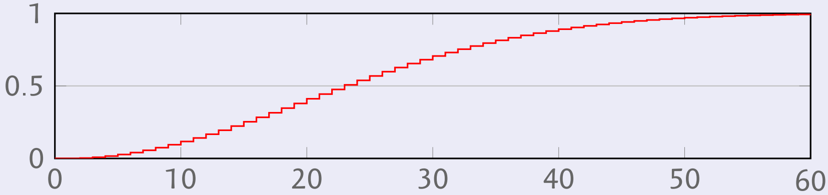
Generic attacks against primitives

- Exhaustive search with small key size
  - Broken: MIFARE Crypto-1 (48 bits), DES (56 bits), A5/1 (64 bits), KeeLoq (64 bits)
- Collisions with small state size
  - Broken: A5/1, MD5

The Birthday Paradox

The birthday paradox

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



Birthday attacks

- With random n-bit strings, first collision after roughly $2^{n/2}$ draws.
- More generally, 2^{2t-n} collisions with 2^t draws

Effect of the state size

Hash function

- Collision attacks with time complexity $2^{n/2}$
- We typically use $n = 256$, $n \geq 128$ for lightweight

Stream cipher

- Time-Memory trade-off with $2^{n/2}$ time and data [Babbage '85, Golic '87]
- We typically use $n \geq 256$, $n \geq 160$ for lightweight

Block cipher

- Good block ciphers secure up to 2^n data
- We typically use $n = 128$, $n = 64$ for lightweight

Today's talk

Modes of operation

- Block ciphers are not used by themselves
- They need a mode of operation: CBC, CTR, CBC-MAC, GCM, ...
  - To achieve a security goal: confidentiality, integrity, authenticated encryption, ...
  - To process several messages with the same key (different IV)
  - To process messages with multiple blocks

Block size is an important security parameter

- Common modes have issues after $2^{n/2}$ blocks of data
  - Security of mode is lower than security of cipher
- Lightweight block ciphers typically use a block size $n = 64$ bits
  - With $n = 64$, the bound is only 32 GB
- How bad is it really?

Today's talk

Modes of operation

- Block ciphers are not used by themselves
- They need a mode of operation: CBC, CTR, CBC-MAC, GCM, ...
  - To achieve a security goal: confidentiality, integrity, authenticated encryption, ...
  - To process several messages with the same key (different IV)
  - To process messages with multiple blocks

Block size is an important security parameter

- Common modes have issues after $2^{n/2}$ blocks of data
  - Security of mode is lower than security of cipher
- Lightweight block ciphers typically use a block size $n = 64$ bits
  - With $n = 64$, the bound is only 32 GB
- How bad is it really?

Security of modes of operations

- To reduce the number of assumptions, study the block cipher and the mode independently

1. Cryptanalysis of the block cipher
   - Try to show non-random behavior
   - After some time, build confidence in the block-cipher
2. Security proof for the mode
   - Assume that the block cipher is good, prove that the mode is good
   - Lower bound on the security of the mode
3. Generic attacks for the mode
   - Attack that work for any choice of the block cipher
   - Upper bound on the security of the mode

Security of modes of operations

- To reduce the number of assumptions,
  study the block cipher and the mode independently

1. Cryptanalysis of the block cipher
   - Try to show non-random behavior
   - After some time, build confidence in the block-cipher
2. Security proof for the mode
   - Assume that the block cipher is good, prove that the mode is good
   - Lower bound on the security of the mode
3. Generic attacks for the mode
   - Attack that work for any choice of the block cipher
   - Upper bound on the security of the mode

Security proofs

- If E is a good PRF, CTR key-stream is indistinguishable from random
- If the key-stream is random, this is a one-time-pad

$$\mathsf{Adv}^{\mathsf{CPA}}_{\mathsf{CTR}\text{-}E}(\sigma) \leq \mathsf{Adv}^{\mathsf{PRF}}_{E}(\sigma)$$

with σ the total number of blocks

- A block-cipher is actually a permutation... PRP/PRF switching lemma

$$\mathsf{Adv}^{\mathsf{PRF}}_{E}(\sigma) \leq \mathsf{Adv}^{\mathsf{PRP}}_{E}(\sigma) + \frac{\sigma^2}{2^n}$$

- The CPA security of CTR is essentially the PRP security of E (the block cipher)
  - As long as the number of encrypted blocks $\sigma \lll 2^{n/2}$
  - Similar results for other modes (CBC, GCM, ...)

Different points of view

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 (circa 2016)

TLS libraries, web browsers no rekeying

OpenVPN no rekeying (PSK mode) / rekey every hour (TLS mode)

Different points of view

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 (circa 2016)

TLS libraries, web browsers no rekeying

OpenVPN no rekeying (PSK mode) / rekey every hour (TLS mode)

Different points of view

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.

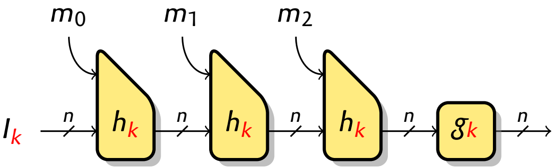
What implementation do (circa 2016)

TLS libraries, web browsers no rekeying

OpenVPN no rekeying (PSK mode) / rekey every hour (TLS mode)

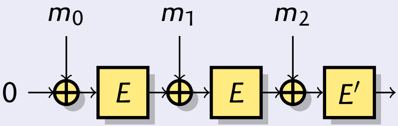
Outline

Example: Iterated Deterministic MACs

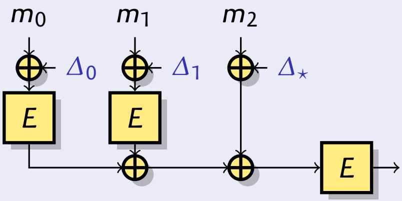


- Many MACs are deterministic iterated constructions
  - BC based: CBC-MAC, PMAC
  - Hash-based: HMAC

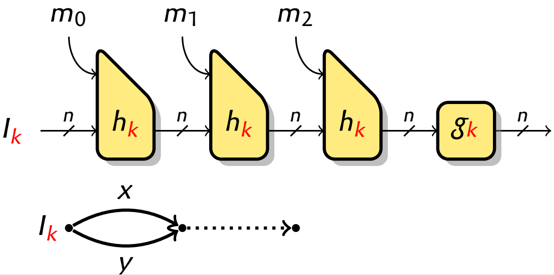
CBC-MAC



PMAC



Example: Iterated Deterministic MACs

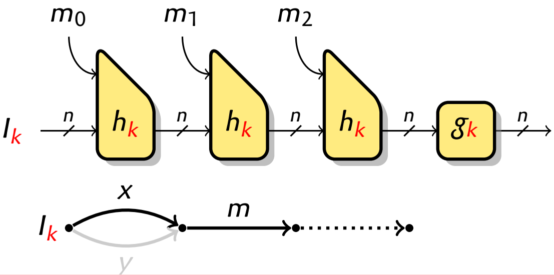


- Many MACs are deterministic iterated constructions
  - BC based: CBC-MAC, PMAC
  - Hash-based: HMAC

Generic attack [Preneel & van Oorschot '95]

1. Find internal collisions $\text{MAC}(x) = \text{MAC}(y)$
   - Query $2^{n/2}$ random short messages
   - 1 internal collision expected, detected in the output
2. Query $t = \text{MAC}(x \| m)$
3. $(y \| m, t)$ is a forgery

Example: Iterated Deterministic MACs

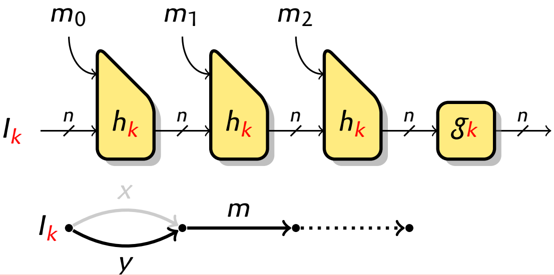


- Many MACs are deterministic iterated constructions
  - BC based: CBC-MAC, PMAC
  - Hash-based: HMAC

Generic attack *[Preneel & van Oorschot '95]*

1. Find internal collisions $\mathrm{MAC}(x) = \mathrm{MAC}(y)$
   - Query $2^{n/2}$ random short messages
   - 1 internal collision expected, detected in the output
2. Query $t = \mathrm{MAC}(x \| m)$
3. $(y \| m, t)$ is a forgery

Example: Iterated Deterministic MACs



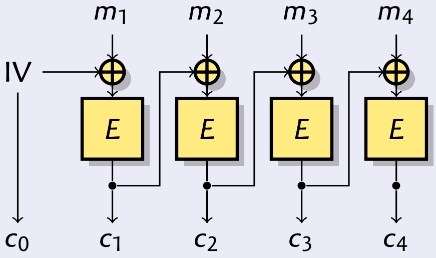
- Many MACs are deterministic iterated constructions
  - BC based: CBC-MAC, PMAC
  - Hash-based: HMAC

Generic attack [Preneel & van Oorschot '95]

1. Find internal collisions $\text{MAC}(x) = \text{MAC}(y)$
   - Query $2^{n/2}$ random short messages
   - 1 internal collision expected, detected in the output
2. Query $t = \text{MAC}(x \| m)$
3. $(y \| m, t)$ is a forgery

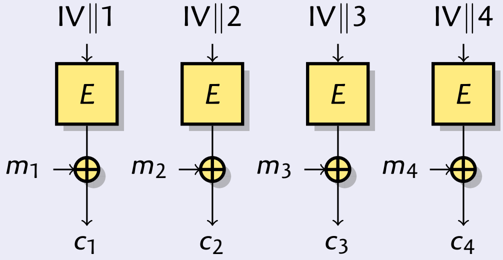
Encryption modes: CBC and CTR

CBC mode



- Security proof up to the birthday bound

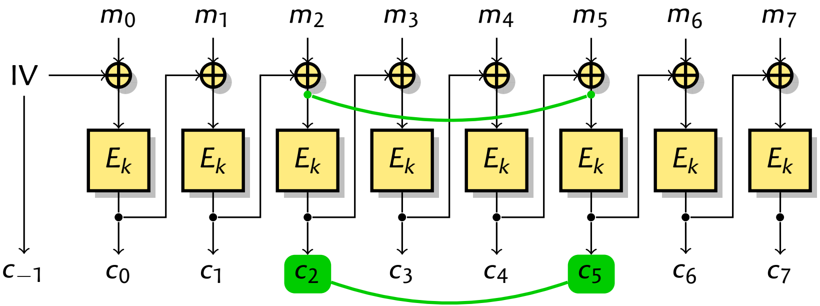
CTR mode



- Security proof up to the birthday bound

CBC collisions

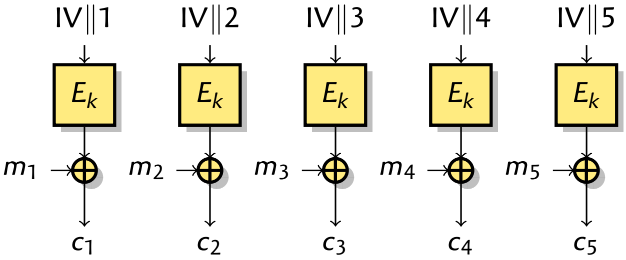
- Well known collision attack against CBC



- 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 distinguishing on CTR

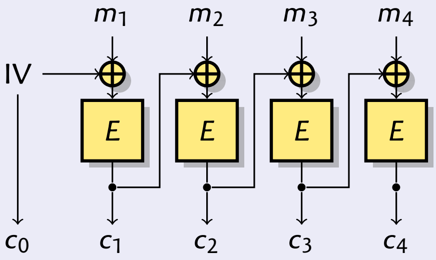
- Well known distinguisher against CTR



- All block cipher input are distinct
- For all $i \neq j$, $m_i \oplus c_i \neq m_j \oplus c_j$
  - Hard to extract plaintext information from inequalities
- Distinguisher: no collisions in $m_i \oplus c_i$
  - Collisions after $2^{n/2}$ blocks with random ciphertext

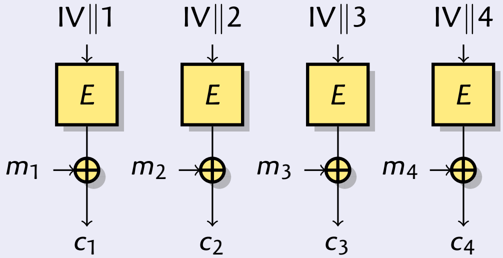
CBC vs. CTR

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: Keystream doesn't collide

Impact

- How bad is it?
  - CBC only leaks xors of a few blocks of plaintexts...
  - CTR doesn't even leak that!

  - Can this leakage be exploited?
  - Do applications encrypt enough data under the same key?

Cryptography engineering *[Ferguson, Schneier, Kohno]*

CTR leaks very little data. [...] It would be reasonable to limit the cipher mode to 2^{60} blocks, which allows you to encrypt 2^{64} bytes but restricts the leakage to a small fraction of a bit. When using CBC mode you should be a bit more restrictive. [...] We suggest limiting CBC encryption to 2^{32} blocks or so.

(talking about a 128-bit block cipher)

Outline

Introduction

Birthday attacks

Exploiting CBC collisions

Plaintext recovery on CTR

Beyond-birthday security

Conclusion

Towards a Practical attack

- Assume a fixed message is repeatedly encrypted (under a fixed key)
  - Including a high value secret (cookie, password, ...) — a few blocks
  - And some known/predictable sections (headers, ...) — 2^t blocks
- Each collision reveals the xor of two plaintext blocks
- With some luck, xor of a known value and the secret

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

- Success after roughly 2^t collisions
  - $2^{n/2-t/2}$ message copies, $2^{n/2+t/2}$ blocks
  - Tradeoff between number of copies and total amount of data
- If rekeying after roughly $2^{n/2}$ blocks, attack still possible
  - $2^{n/2}$ message copies, $2^{n/2+t}$ blocks

Towards a Practical attack

2^t

$2^{n-t/2}$ 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

Towards a Practical 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
$2^{n-t/2}$	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

Towards a Practical 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

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

Towards a Practical 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
$2^{n-t/2}$	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

Towards a Practical attack

2^t

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

Towards a Practical attack

2^t (width of the grid)

$2^{n-t/2}$ Ciphertexts (height of the grid)

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

Towards a Practical 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
$2^{n-t/2}$	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

Towards a Practical 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
$2^{n-t/2}$	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

Towards a Practical 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

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

Towards a Practical 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
$2^{n-t/2}$	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

Towards a Practical attack

2^t

Plaintext; $2^{n-t/2}$ 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

Towards a Practical attack

2^t

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

Towards a Practical attack

2^t

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

Towards a Practical attack

- Assume a fixed message is repeatedly encrypted (under a fixed key)
  - Including a high value secret (cookie, password, ...) — a few blocks
  - And some known/predictable sections (headers, ...) — 2^t blocks
- Each collision reveals the xor of two plaintext blocks
- With some luck, xor of a known value and the secret

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

- Success after roughly 2^t collisions
  - $2^{n/2-t/2}$ message copies, $2^{n/2+t/2}$ blocks
  - Tradeoff between number of copies and total amount of data
- If rekeying after roughly $2^{n/2}$ blocks, attack still possible
  - $2^{n/2}$ message copies, $2^{n/2+t}$ blocks

Towards a Practical attack

- Assume a fixed message is repeatedly encrypted (under a fixed key)
  - Including a high value secret (cookie, password, ...) — a few blocks
  - And some known/predictable sections (headers, ...) — 2^t blocks
- Each collision reveals the xor of two plaintext blocks
- With some luck, xor of a known value and the secret

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

- Success after roughly 2^t collisions
  - $2^{n/2-t/2}$ message copies, $2^{n/2+t/2}$ blocks
  - Tradeoff between number of copies and total amount of data
- If rekeying after roughly $2^{n/2}$ blocks, attack still possible
  - $2^{n/2}$ message copies, $2^{n/2+t}$ blocks

HTTPS encryption: HTTP over TLS

HTTP

- Hypertext Transfer Protocol
  - Request/response (text)
  - Headers and body

```
GET /index.html HTTP/1.1
User-Agent: Firefox
```

```
HTTP/1.1 200 OK
Content-Type: text/html

<html>
  <body>...
```

TLS

- Transport Layer Security
  - Evolution of Netscape's SSL
  - Current version: TLS 1.2
- Stream encryption protocol
  - Algorithm negotiation
  - Handshake: asym. crypto
  - Transport: sym. crypto
- Each HTTP message encrypted in a TLS packet

64-bit block ciphers in HTTPS

- 3DES is one of the ciphers supported in TLS
  - Mandatory to implement up to TLS 1.1

3DES usage

- About 1% of HTTPS connections use 3DES
  - Outdated client/servers
    - Windows XP / Windows 2003 Server don't support AES out of the box
  - Many poorly configured servers support AES, but prefer 3DES

Session length

- HTTP 1.1 allows connection reuse (`Keep-alive`)
- *Web browsers* reuse a connection as long as possible
- *Web servers:* Apache, Nginx limit to 200 queries per session
  - In practice, many high-profile website support very long sessions

HTTP authentication tokens

- HTTP is stateless: authentication tokens sent with every request
  - HTTP 1.1 Keep-alive sends many requests in the same connection

HTTP Basic Auth (RFC 7617)

- User/Password sent in a header (base64 encoded)

```
Authorization: Basic dGVzdDoxMjPCow=
```

HTTP Cookies (RFC 6265)

1. User sends password in a from
2. Server reply with a Cookie
3. Cookie is included in every subsequent request

```
Cookie: C=123456
```

Javascript attack

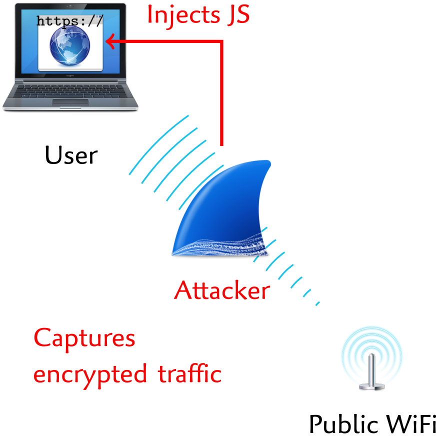
- A webpage is not just data, it includes code
- Malicious website can send requests to third party
- Requests include authentication cookies

Javascript attack

```
var url = "https://www.facebook.com/index.html";
var xhr = new XMLHttpRequest;

while(true) {
    xhr.open("HEAD", url, false);
    xhr.withCredentials = true;
    xhr.send();
    xhr.abort();
}
```

BEAST Attack Setting [Duong & Rizzo 2011]



- Attacker has access to the network (*eg.* public WiFi)
- User logged-in to secure website (w/ cookie or BasicAuth)

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

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

Disclosure and mitigation

Sweet32 attack

- Birthday attacks against 64-bit block ciphers are practical

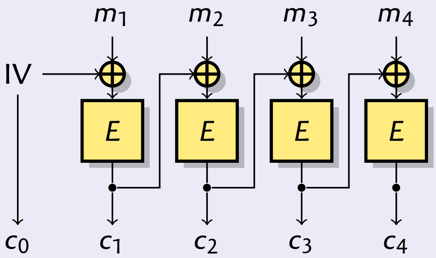
On the Practical (In-)Security of 64-bit Block Ciphers
Karthikeyan Bhargavan, G. L. [ACM CCS '16]

- OpenVPN 2.4 has cipher negotiation defaulting to AES
- Mozilla has implemented data limits in Firefox 51 (1M records)
- NIST has limited 3DES usage to 2^{20} blocks per key
- OpenSSL has updated the list of `HIGH` security ciphers (sorted)
  - Before 2014: `AES256, CAMELLIA256, 3DES, AES128, CAMELLIA128`
  - After 2014: `AES256, CAMELLIA256, AES128, CAMELLIA128, 3DES`
  - After 2016: `AES256, CAMELLIA256, AES128, CAMELLIA128`

Outline

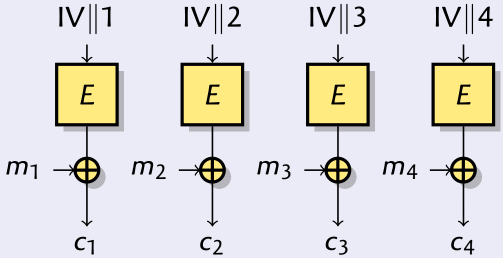
CBC vs. CTR

CBC mode



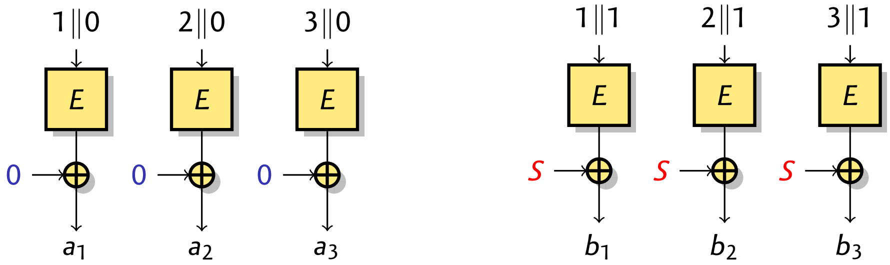
- Security proof up to the birthday bound
- Collisions reveals xor of two plaintext blocks

CTR mode



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

Plaintext recovery on CTR



Plaintext recovery

- Collect two kind of blocks
  - $a_i = E(i)$
  - $b_j = E(j) \oplus S$
- $\forall i, j,\ S \neq a_i \oplus b_j$

The missing difference problem

- Given $\mathcal{A}$ and $\mathcal{B}$, and a hint $\mathcal{S}$
- Find $S \in \mathcal{S}$ such that:

$$\forall (a, b) \in \mathcal{A} \times \mathcal{B},\ S \neq a \oplus b\ .$$

Missing difference problem algorithms

Algorithms for the missing difference problem

Sieving	Complexity $\tilde{\mathcal{O}}(2^n)$	[McGrew]
Searching	Complexity $\tilde{\mathcal{O}}(2^{n/2}\sqrt{\lvert\mathcal{S}\rvert})$	[McGrew]
Known-prefix sieving	Complexity $\tilde{\mathcal{O}}(2^{n/2} + 2^{\dim\langle\mathcal{S}\rangle})$	[New]
Fast convolution sieving	Complexity $\tilde{\mathcal{O}}(2^{2n/3})$	[New]

The Missing Difference Problem, and its Applications to Counter Mode Encryption
Ferdinand Sibleyras, G. L. [Eurocrypt '18]

- Plaintext recovery with birthday complexity
- CTR not more secure than CBC

Application to CTR (CPSS queries)

- Plaintext recovery using the known-prefix sieving algorithm
- Two kind of queries:

  Queries Q_1 with half-block header

H_1	S_1	S_2	S_3	S_4

  Queries Q_2 with full-block header

H_1	H_2	S_1	S_2	S_3	S_4

1. Recover S_1 using the first block of each query:
   $\mathcal{A} = \{\mathcal{E}(H_1 \| H_2)\}, \mathcal{B} = \{\mathcal{E}(H_1 \| S_1)\}$. → Missing difference: $0 \| (S_1 \oplus H_2)$.
2. When S_1 is known, recover S_2, with the first and second blocks of Q_2 queries:
   $\mathcal{A} = \{\mathcal{E}(H_1 \| H_2)\}, \mathcal{B} = \{\mathcal{E}(S_1 \| S_2)\}$. → Missing difference: $(S_1 \oplus H_1) \| (S_2 \oplus H_2)$.
3. When S_2 is known, recover S_3:
   $\mathcal{A} = \{\mathcal{E}(H_1 \| H_2)\}, \mathcal{B} = \{\mathcal{E}(S_2 \| S_3)\}$. → Missing difference: $(S_2 \oplus H_1) \| (S_3 \oplus H_2)$.
4. ...

Application to CTR (CPSS queries)

- Plaintext recovery using the known-prefix sieving algorithm
- Two kind of queries:

  Queries Q_1 with half-block header

H_1 S_1	S_2 S_3	S_4

  Queries Q_2 with full-block header

H_1 H_2	S_1 S_2	S_3 S_4

1. Recover S_1 using the first block of each query:
   $\mathcal{A} = \{\mathcal{E}(H_1 \| H_2)\}, \mathcal{B} = \{\mathcal{E}(H_1 \| S_1)\}.$ $\rightarrow$ Missing difference: $0 \| (S_1 \oplus H_2).$
2. When S_1 is known, recover S_2, with the first and second blocks of Q_2 queries:
   $\mathcal{A} = \{\mathcal{E}(H_1 \| H_2)\}, \mathcal{B} = \{\mathcal{E}(S_1 \| S_2)\}.$ $\rightarrow$ Missing difference: $(S_1 \oplus H_1) \| (S_2 \oplus H_2).$
3. When S_2 is known, recover S_3:
   $\mathcal{A} = \{\mathcal{E}(H_1 \| H_2)\}, \mathcal{B} = \{\mathcal{E}(S_2 \| S_3)\}.$ $\rightarrow$ Missing difference: $(S_2 \oplus H_1) \| (S_3 \oplus H_2).$
4. ...

Application to CTR (CPSS queries)

- Plaintext recovery using the known-prefix sieving algorithm
- Two kind of queries:

Queries Q_1 with half-block header

H_1	S_1	S_2	S_3	S_4

Queries Q_2 with full-block header

H_1	H_2	S_1	S_2	S_3	S_4

1. Recover S_1 using the first block of each query:
   $\mathcal{A} = \{\mathcal{E}(H_1 \| H_2)\}, \mathcal{B} = \{\mathcal{E}(H_1 \| S_1)\}$. $\to$ Missing difference: $0 \| (S_1 \oplus H_2)$.
2. When S_1 is known, recover S_2, with the first and second blocks of Q_2 queries:
   $\mathcal{A} = \{\mathcal{E}(H_1 \| H_2)\}, \mathcal{B} = \{\mathcal{E}(S_1 \| S_2)\}$. $\to$ Missing difference: $(S_1 \oplus H_1) \| (S_2 \oplus H_2)$.
3. When S_2 is known, recover S_3:
   $\mathcal{A} = \{\mathcal{E}(H_1 \| H_2)\}, \mathcal{B} = \{\mathcal{E}(S_2 \| S_3)\}$. $\to$ Missing difference: $(S_2 \oplus H_1) \| (S_3 \oplus H_2)$.
4. ...

Security of modes of operation

- All common modes have security proofs up to the birthday bound
- Plaintext recovery with one of these techniques
  - Collision attack if collisions happen
  - Missing difference problem if collisions don't happen

Example: f8 mode

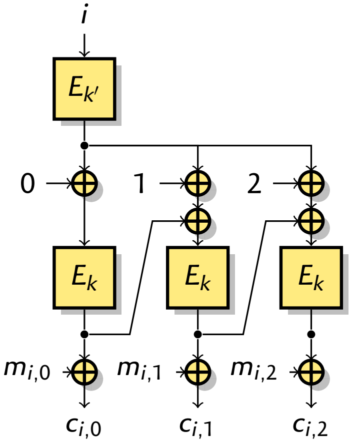
- Used in 3G telephony
- With a 64-bit block cipher (Kasumi)
- Designed to limit birthday attacks
- Missing difference attack
  - First block of keystream does not repeat
  - Instance of missing difference problem

Security of modes of operation

- All common modes have security proofs up to the birthday bound
- Plaintext recovery with one of these techniques
  - Collision attack if collisions happen
  - Missing difference problem if collisions don't happen

Example: f8 mode

- Used in 3G telephony
- With a 64-bit block cipher (Kasumi)
- Designed to limit birthday attacks
- Missing difference attack
  - First block of keystream does not repeat
  - Instance of missing difference problem

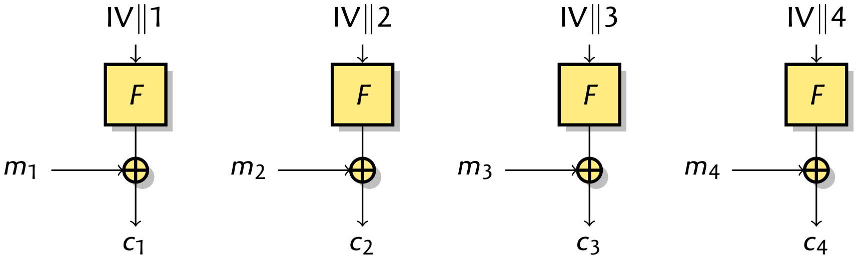


Outline

Countermeasures

1. Use a block cipher with larger block size (*eg* AES, Rinjdael-256)
   - Not lightweight
2. Limit the amount of data per key (rekeying)
   - Often ignored by implementers
     - Adversary can make you generate data
   - Need very low limit with 64-bit blocks
     - NIST now limits 3DES to 2^{20} blocks per key (8MB)
     - NIST lightweight call requires at least 2^{50} blocks per key
   - Rekeying allows multi-key attacks
     - Birthday attack to recover one key out of many
3. Use better modes of operation?
   - Security beyond the birthday bound

Better PRFs

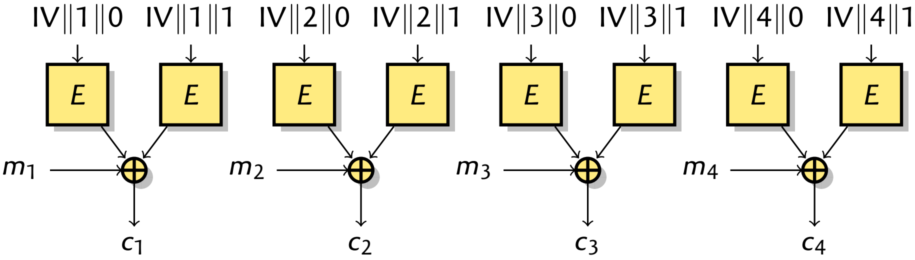


- The security loss of CTR is because of the PRF/PRP switching lemma

$$\mathrm{Adv}^{\mathrm{CPA}}_{\mathrm{CTR}\text{-}E}(\sigma) \leq \mathrm{Adv}^{\mathrm{PRF}}_{F}(\sigma)$$

- We can build a better PRF as $E(0\|x) \oplus E(1\|x)$ (Xor of Permutations)
  - Security close to 2^n [Patarin'08], [Patarin'13], [DHT, Crypto'17]

Better PRFs

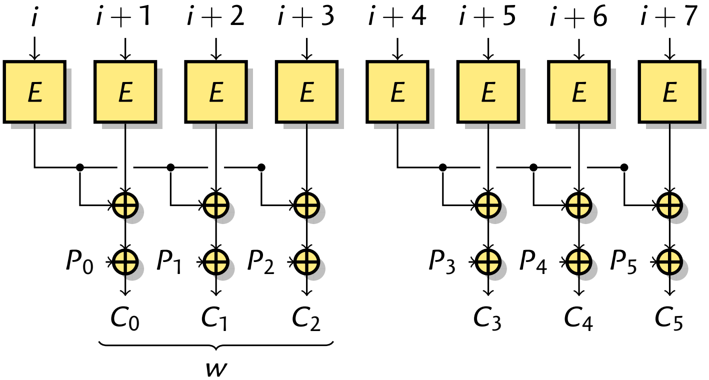


- The security loss of CTR is because of the PRF/PRP switching lemma

$$\mathsf{Adv}^{\mathsf{CPA}}_{\mathsf{CTR}\text{-}E}(\sigma) \leq \mathsf{Adv}^{\mathsf{PRF}}_{F}(\sigma)$$

- We can build a better PRF as $E(0\|x) \oplus E(1\|x)$ (Xor of Permutations)
  - Security close to 2^n [Patarin'08], [Patarin'13], [DHT, Crypto'17]

CENC

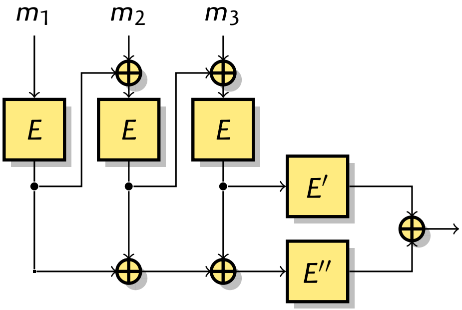


- CENC: Similar security as CTR-XoP with smaller overhead
  - Designed by Iwata, security proof up to $2^{2n/3}$ [FSE '06]
  - Security proof up to $2^n/w$ [Iwata, Mennink & Vizár '16]

BBB secure MACs

- All iterated deterministic MACs are broken by collision attack with 2^n messages

1. Use a larger internal state
   - SUM-ECBC, PMAC+, 3kf9 have a $2n$-bit internal state with an n-bit block cipher
   - Security proofs up to $2^{2n/3}$
   - Open problem: no known attack, what is their actual security?



2. Use a non-deterministic MAC (randomized or IV-based)
   - RMAC, Wegman-Carter: security up to almost 2^n
   - In practice: Wegman-Cater-Shoup birthday security

Wegman-Carter MACs

- Wegman-Carter: build a MAC from a universal hash function and a PRF

$$WC(N, M) = H_{k_1}(M) \oplus F_{k_2}(N).$$

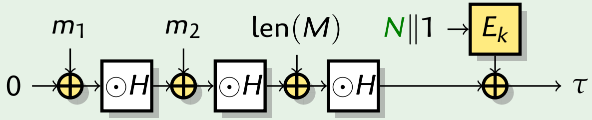
  - Security close to 2^n

- Wegman-Carter-Shoup: use a block cipher as a PRF

$$WCS(N, M) = H_{k_1}(M) \oplus E_{k_2}(N),$$

  - Birthday security

Example: Polynomial-based hasing (GMAC, Poly1305-AES)



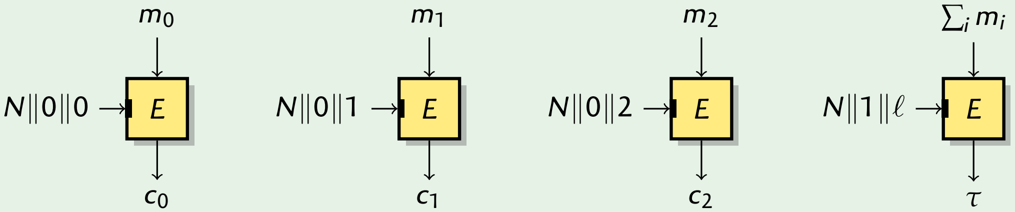
- Better options: WMAC, EWCDM, WC with XoP, ...
  - Security close to 2^n

Using Tweakable block-ciphers

- Another option: use a different primitive

- Tweakable block cipher [Liskov, Rivest & Wagner '02]
  - For each key, a family of independent permutations (indexed by public tweak)
  - Dedicated designs: SCREAM, Deoxys, Joltik, Skinny

TAE/ΘCB: authenticated encryption *[LRW'02, Rogaway'04]*



- Secure up to 2^n blocks with an n-bit state

Conclusion

- Security of modes is lower than security of block ciphers
- Distinguishers matter!
  - All classical modes broken with collisions or missing differences
  - Plaintext recovery possible with birthday complexity

Security issues with small block sizes

- Practical attacks against 64-bit block cipher with classical modes
- Be careful with 64-bit lightweight block ciphers...
- More research needed on lightweight modes, in addition to lightweight bloc ciphers