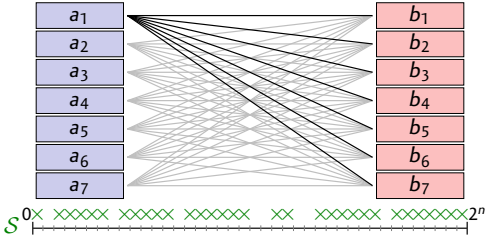
The Missing Difference Problem

Gaëtan Leurent, Ferdinand Sibleyras

Inria, France

Flexible Symmetric Cryptography Workshop



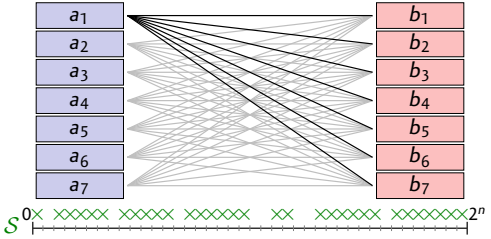
The Missing Difference Problem

And its Applications to Counter Mode Encryption

Gaëtan Leurent, Ferdinand Sibleyras

Inria, France

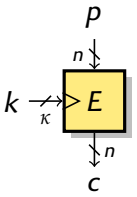
Flexible Symmetric Cryptography Workshop



Block ciphers and Modes of operation

- A block cipher is a family of permutations:

$$\begin{array}{ccccc} \{0,1\}^\kappa, & \{0,1\}^n & \to & \{0,1\}^n \\ k & , \quad p & \mapsto & c \end{array}$$

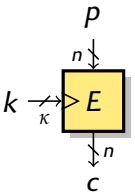


- It is used with a mode of operation: CBC, CTR, GCM, ...
  - To encrypt several messages with the same key (different IV)
  - To process messages with multiple blocks
  - Important example: CTR

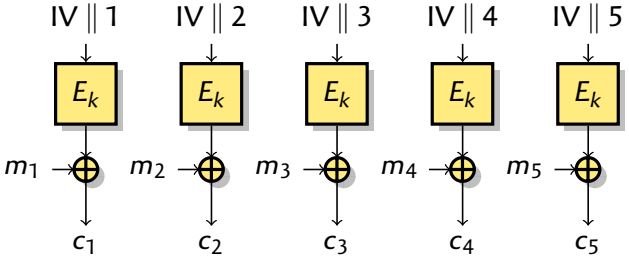
Block ciphers and Modes of operation

- A block cipher is a family of permutations:

$$\{0,1\}^\kappa, \{0,1\}^n \to \{0,1\}^n$$
$$k \quad , \quad p \quad \mapsto c$$



- It is used with a mode of operation: CBC, CTR, GCM, ...
  - To encrypt several messages with the same key (different IV)
  - To process messages with multiple blocks
  - Important example: CTR



Security of modes of operation

- 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, ...)

Security of modes of operation

- 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, ...)

Security of modes of operation

- 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, ...)

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 (circa 2016)

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 (circa 2016)

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.

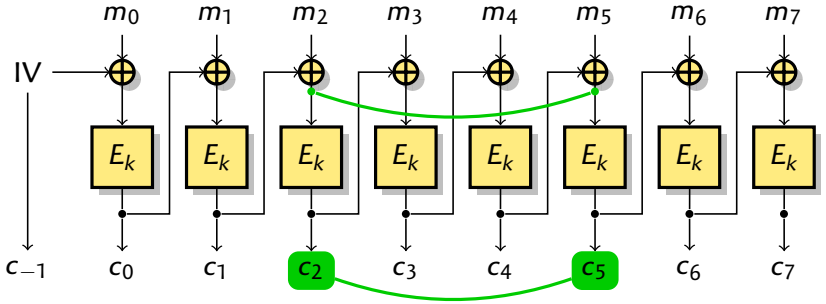
What implementation do (circa 2016)

TLS libraries, web browsers no rekeying

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

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

Security of CBC in practice

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

Sweet32 attack

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

- CBC is still used with 64-bit block-ciphers
  - 1—2 % of HTTPS connections used 3DES
  - OpenVPN used Blowfish by default
- Limit at 2^{32} blocks: 32GB (not suitable for NIST lightweight competition)
  - Rekeying just before 2^{32} blocks doesn't help much
- Practical attack in web context (BEAST setting)
  - Expected complexity: 38 hours for 785 GB (tradeoff query size / number of query).

Security of CBC in practice

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

Sweet32 attack

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

- CBC is still used with 64-bit block-ciphers
  - 1—2 % of HTTPS connections used 3DES
  - OpenVPN used Blowfish by default
- Limit at 2^{32} blocks: 32GB (not suitable for NIST lightweight competition)
  - Rekeying just before 2^{32} blocks doesn't help much
- Practical attack in web context (BEAST setting)
  - Expected complexity: 38 hours for 785 GB (tradeoff query size / number of query).

Security of CBC in practice

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

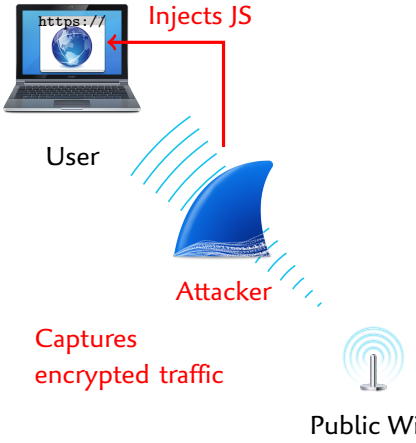
Sweet32 attack

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

- CBC is still used with 64-bit block-ciphers
  - 1—2 % of HTTPS connections used 3DES
  - OpenVPN used Blowfish by default
- Limit at 2^{32} blocks: 32GB (not suitable for NIST lightweight competition)
  - Rekeying just before 2^{32} blocks doesn't help much
- Practical attack in web context (BEAST setting)
  - Expected complexity: 38 hours for 785 GB (tradeoff query size / number of query).

BEAST Attack Setting

[Duong & Rizzo 2011]



- Attacker has access to the network (public WiFi)
- User logged-in to secure website w/ cookie

1. Attacker uses JS to generate traffic
   - Tricks victim to malicious site
   - JS makes *cross-origin* requests
   - Every request includes cookie
2. Attacker captures encrypted data

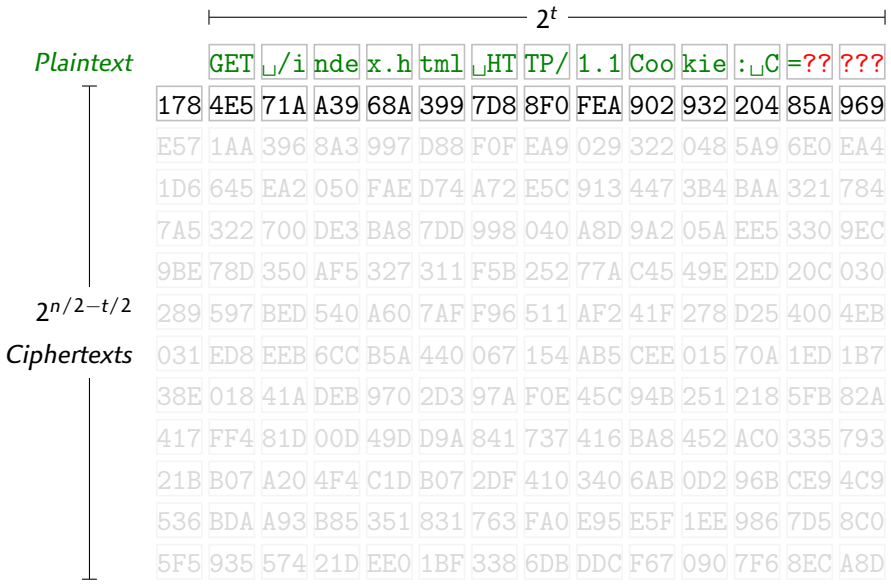
- Chosen-plaintext attack
- Can be modeled as Chosen-Prefix Secret-Suffix: $M \mapsto \mathcal{E}(M \,\|\, S)$ [Hoang & *al.*, Crypto'15]

Attack in practice: Sweet32

[Bhargavan & L, CCS'16]

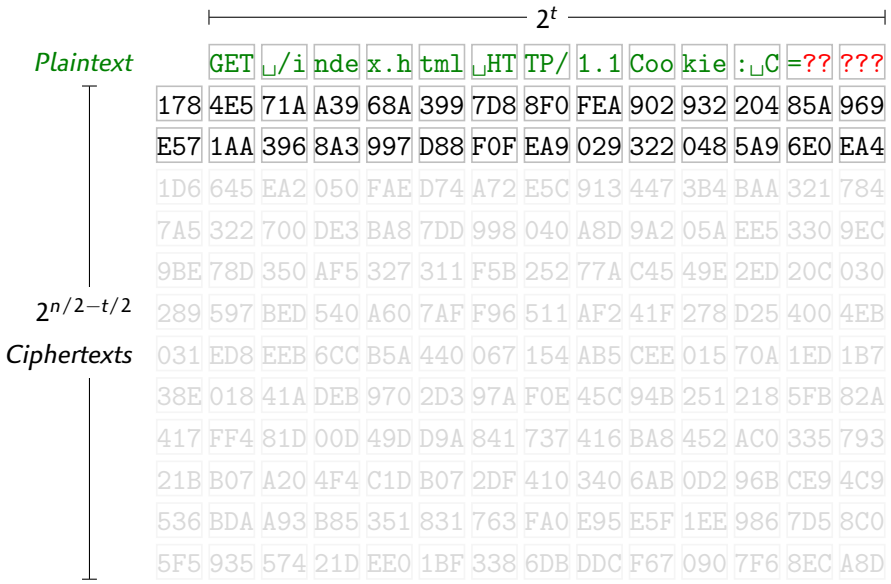
Attack in practice: Sweet32

[Bhargavan & L, CCS'16]



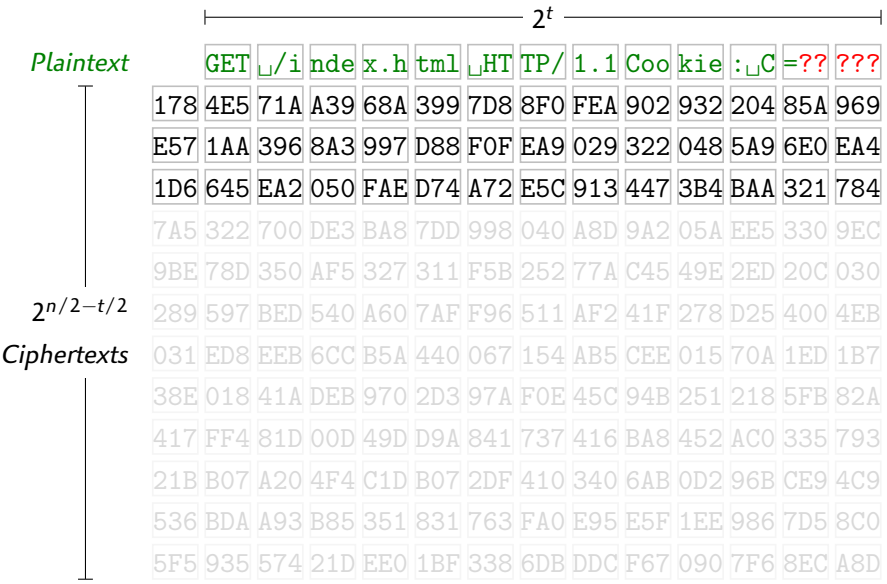
Attack in practice: Sweet32

[Bhargavan & L, CCS'16]

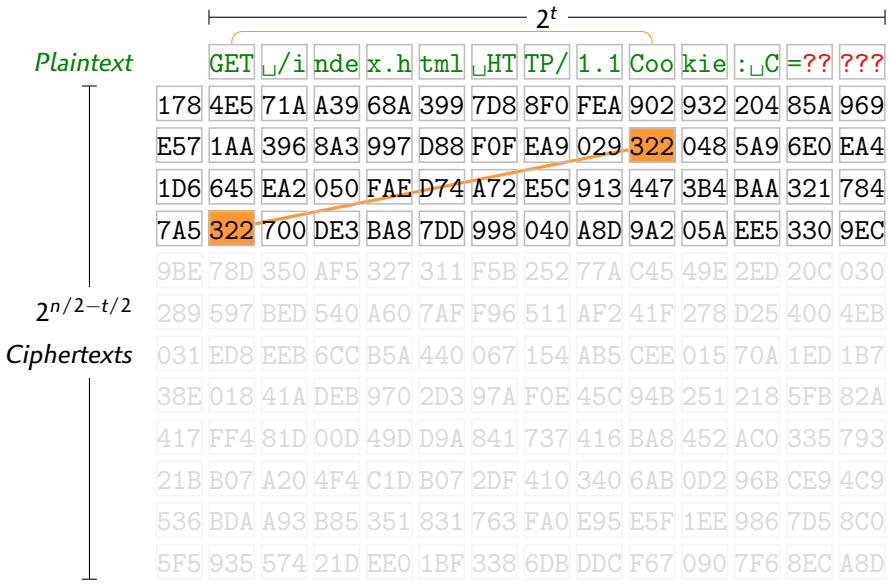


Attack in practice: Sweet32

[Bhargavan & L, CCS'16]

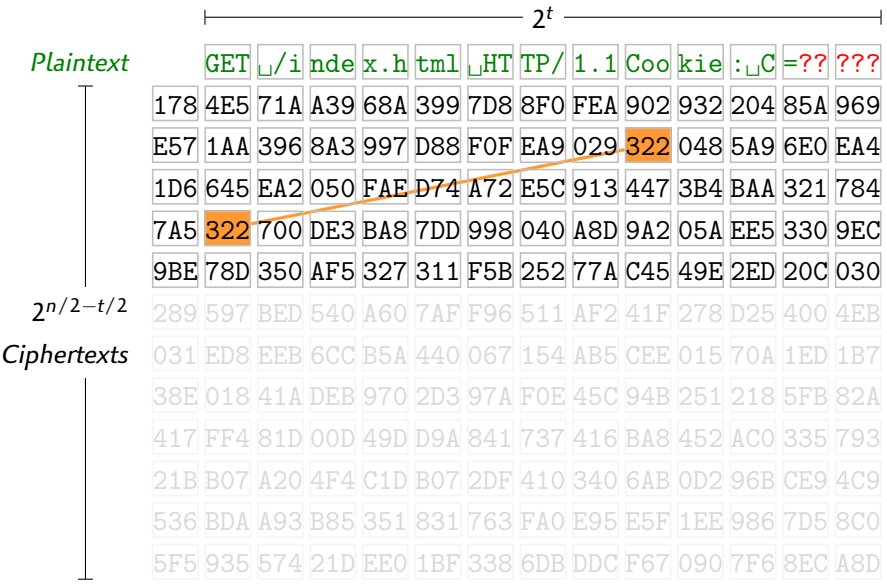


Attack in practice: Sweet32 [Bhargavan & L, CCS'16]



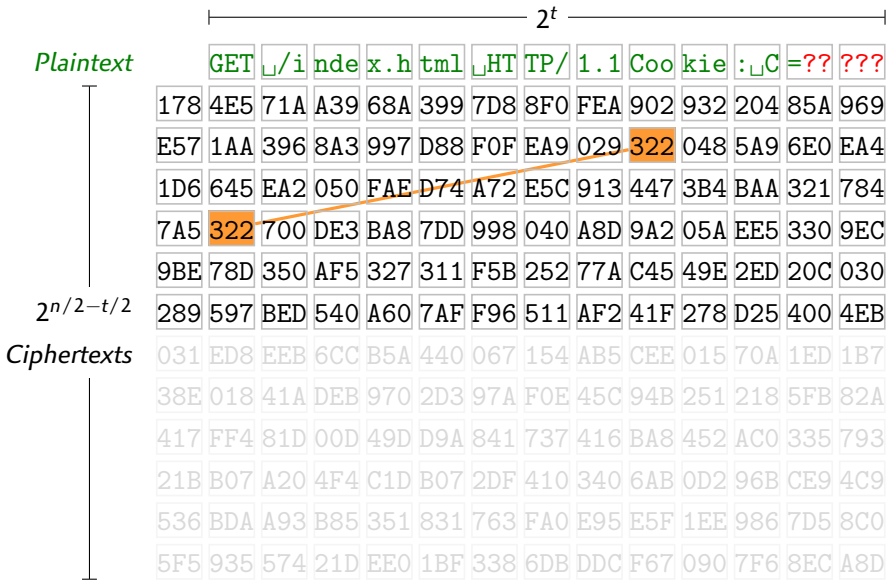
Attack in practice: Sweet32

[Bhargavan & L, CCS'16]



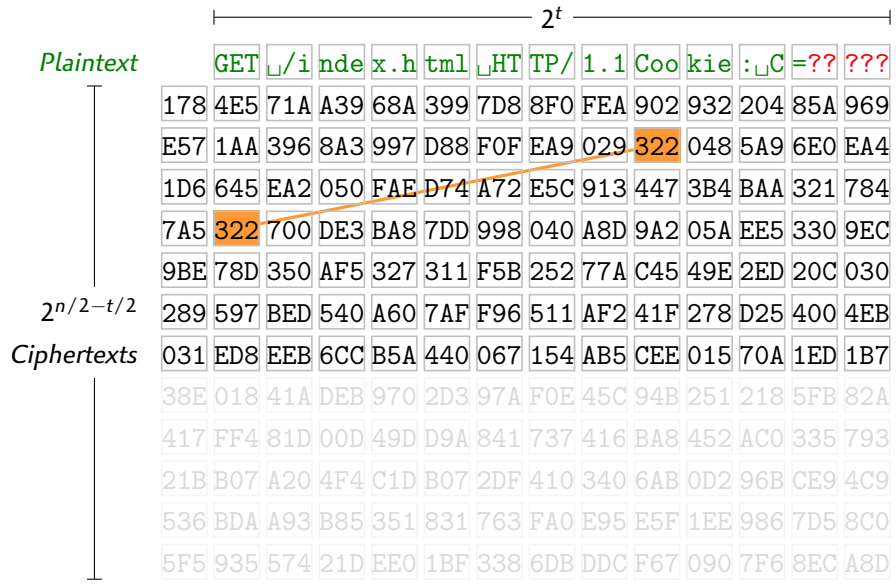
Attack in practice: Sweet32

[Bhargavan & L, CCS'16]



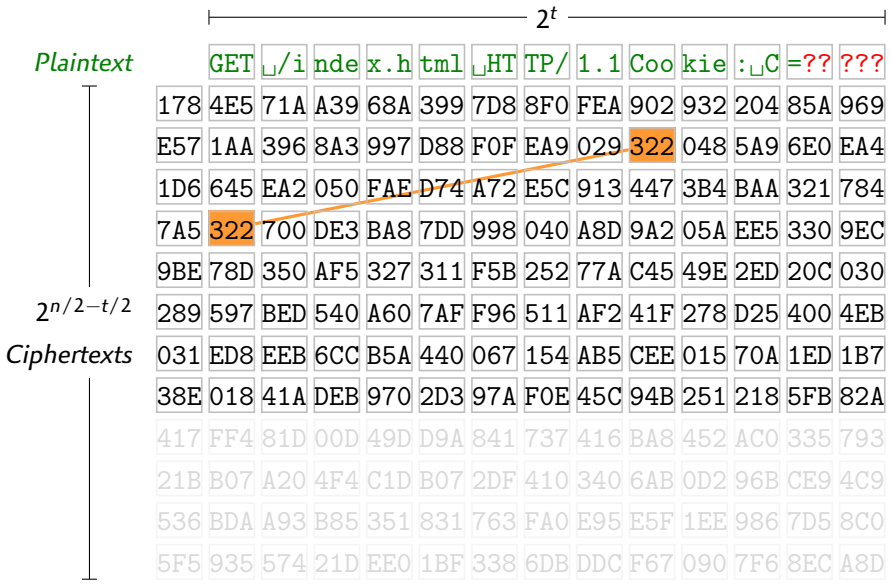
Attack in practice: Sweet32

[Bhargavan & L, CCS'16]



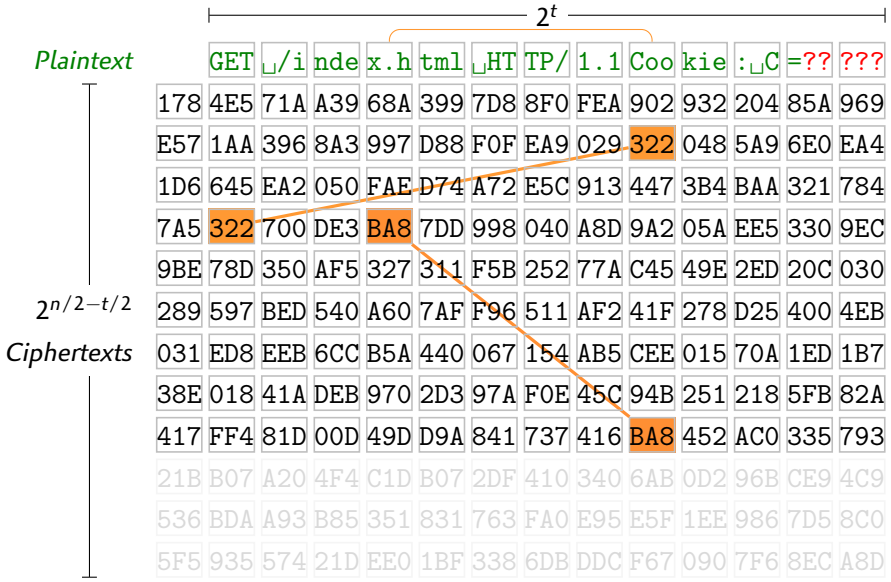
Attack in practice: Sweet32

[Bhargavan & L, CCS'16]



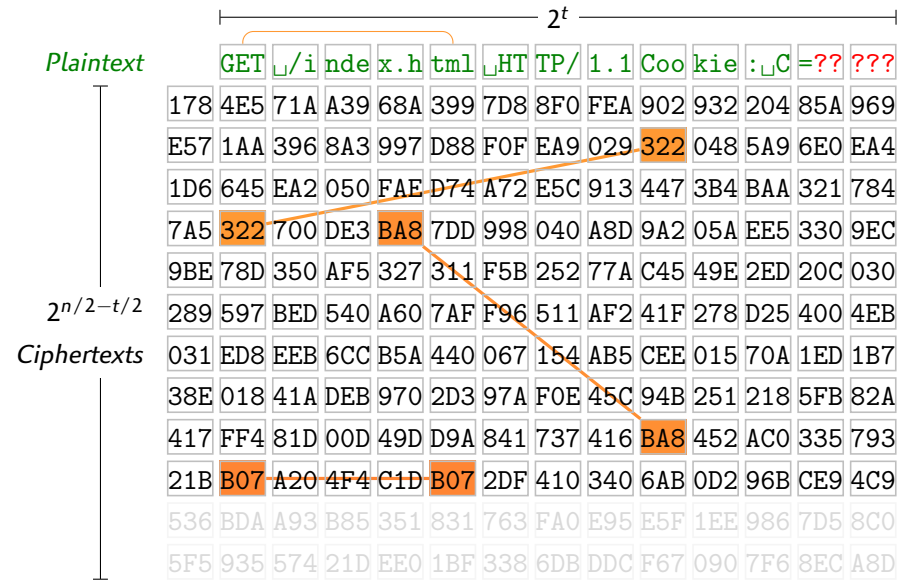
Attack in practice: Sweet32

[Bhargavan & L, CCS'16]



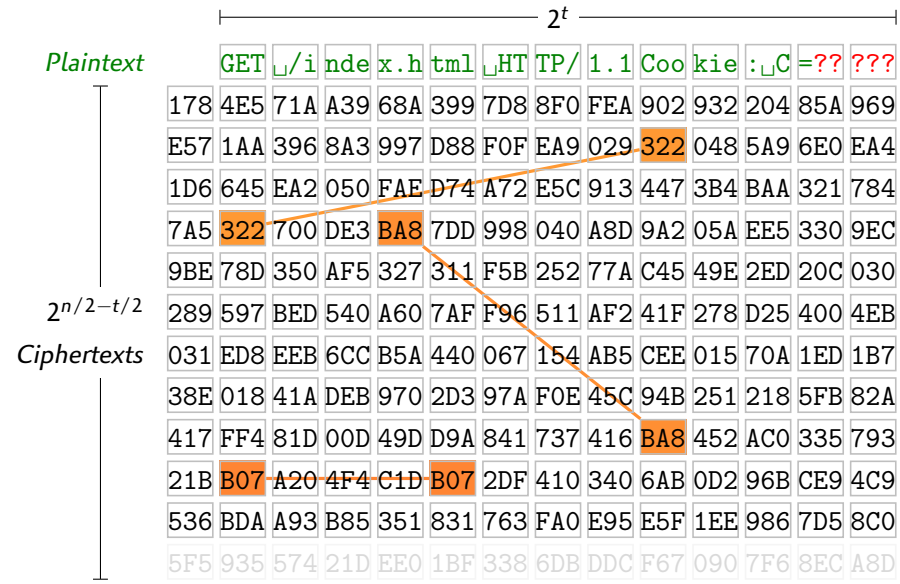
Attack in practice: Sweet32

[Bhargavan & L, CCS'16]



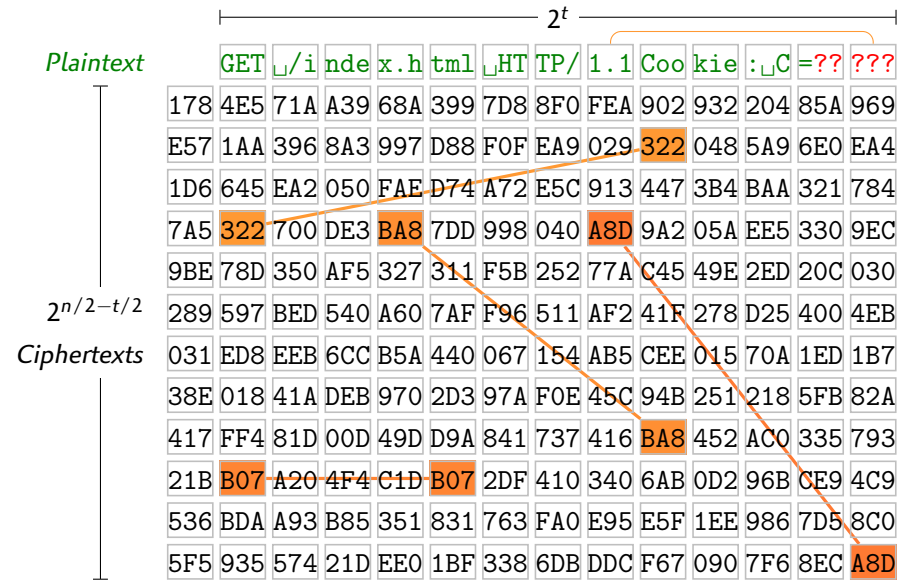
Attack in practice: Sweet32

[Bhargavan & L, CCS'16]



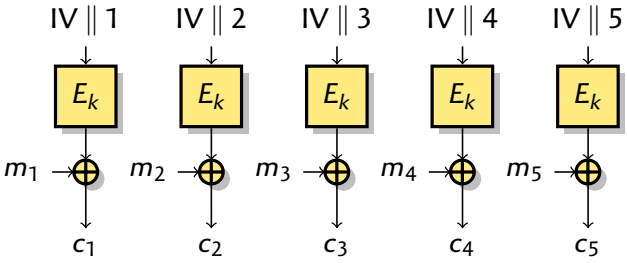
Attack in practice: Sweet32

[Bhargavan & L, CCS'16]



Birthday distinguishing on CTR

- Well known distinguisher against CTR

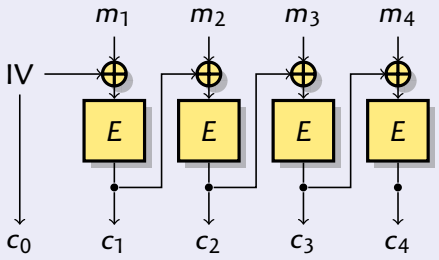


- All block cipher input are distinct
- For all $i \neq j$, $m_i \oplus c_i \neq m_j \oplus c_j$
  - Hard to extract plaintext information from inequalities
- Distinguisher: collision after $2^{n/2}$ blocks with random ciphertext

CBC vs. CTR

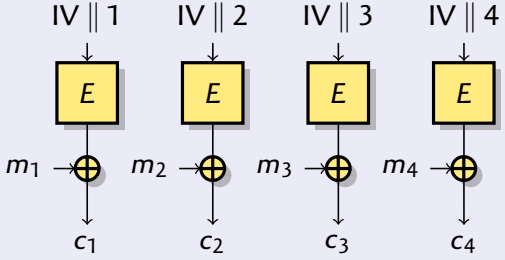
CBC mode

- Collisions reveals
  xor of two plaintext blocks



CTR mode

- Distinguishing attack:
  Key stream doesn't collide



CBC vs. CTR

CBC mode

- Collisions reveals
  xor of two plaintext blocks

CTR mode

- Distinguishing attack:
  Key stream doesn't collide

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.

Overview of this work

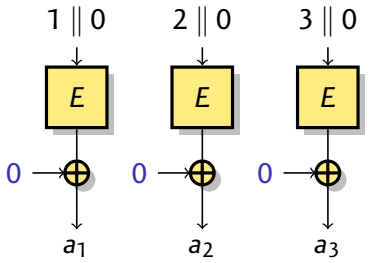
Main question

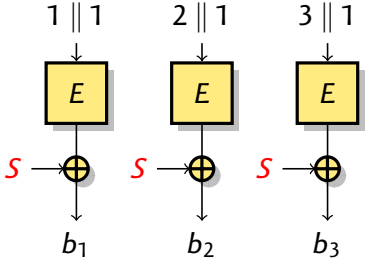
- What is the concrete impact of the CTR distinguisher?
- Can we recover plaintext?

- Assume web adversary (BEAST setting)
  - Can be modeled as CPSS queries: $M \mapsto \mathcal{E}(M \,\|\, S)$ [Hoang & *al.*, Crypto'15]
- Assume that the block cipher is ideal
- Look at full time complexity

- CTR security related to nice algorithm problem: the missing difference problem
  1. New algorithms for the missing difference problem
  2. Application to CTR and Wegman-Carter MACs

Plaintext recovery on CTR





Plaintext recovery

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

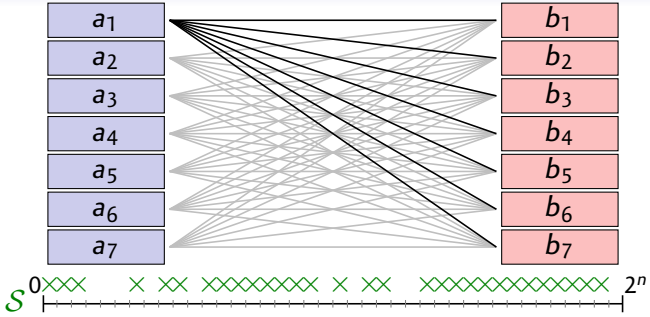
The missing difference problem

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

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

Sieving algorithm

[McGrew, FSE'13]



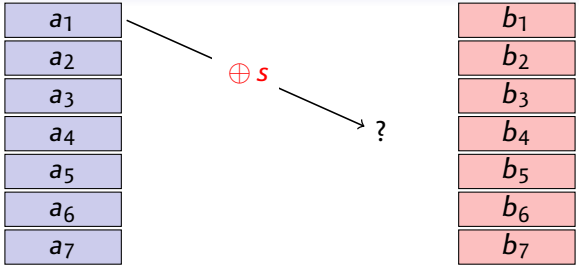
- Compute all $a_i \oplus b_j$, remove from a sieve $\mathcal{S}$

Analysis: Coupon collector problem

- To exclude 2^n candidates S, we need $n \cdot 2^n$ values $a_i \oplus b_j$
  - Lists $\mathcal{A}$ and $\mathcal{B}$ of size $\sqrt{n} \cdot 2^{n/2}$. Complexity: $\tilde{\mathcal{O}}(2^n)$

Searching algorithm

[McGrew, FSE'13]



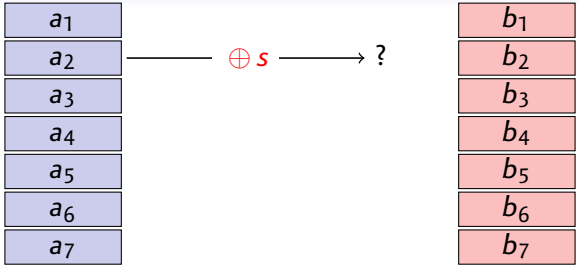
- Make a guess for S, and verify
- Complexity $\tilde{\mathcal{O}}(2^{n/2}\sqrt{|\mathcal{S}|})$ with unbalanced $\mathcal{A}, \mathcal{B}$

Try Guess

for a **in** $\mathcal{A}$ **do**
  if $(s \oplus a) \in \mathcal{B}$ **then**
    return 0
return 1

Searching algorithm

[McGrew, FSE'13]



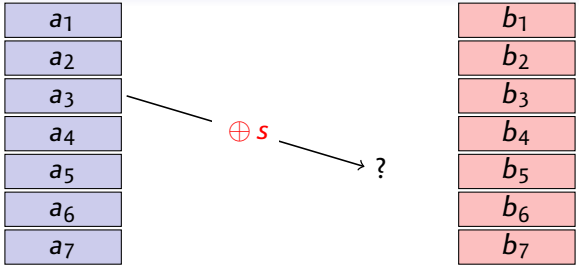
- Make a guess for S, and verify
- Complexity $\tilde{\mathcal{O}}(2^{n/2}\sqrt{|\mathcal{S}|})$ with unbalanced $\mathcal{A}, \mathcal{B}$

Try Guess

for a **in** $\mathcal{A}$ **do**
  if $(s \oplus a) \in \mathcal{B}$ **then**
    return 0
return 1

Searching algorithm

[McGrew, FSE'13]



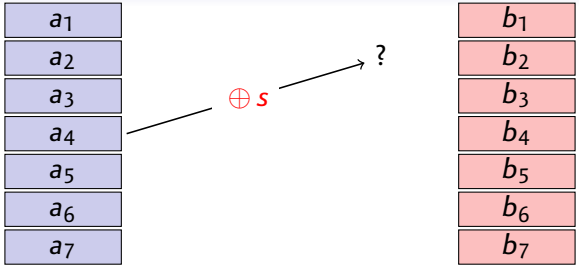
- Make a guess for S, and verify
- Complexity $\tilde{\mathcal{O}}(2^{n/2}\sqrt{|\mathcal{S}|})$ with unbalanced $\mathcal{A}, \mathcal{B}$

Try Guess

for a **in** $\mathcal{A}$ **do**
  if $(s \oplus a) \in \mathcal{B}$ **then**
    return 0
return 1

Searching algorithm

[McGrew, FSE'13]



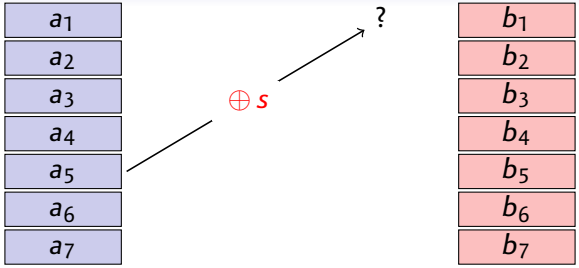
- Make a guess for S, and verify
- Complexity $\tilde{\mathcal{O}}(2^{n/2}\sqrt{|\mathcal{S}|})$ with unbalanced $\mathcal{A}, \mathcal{B}$

Try Guess

for a **in** $\mathcal{A}$ **do**
  if $(s \oplus a) \in \mathcal{B}$ **then**
    return 0
return 1

Searching algorithm

[McGrew, FSE'13]



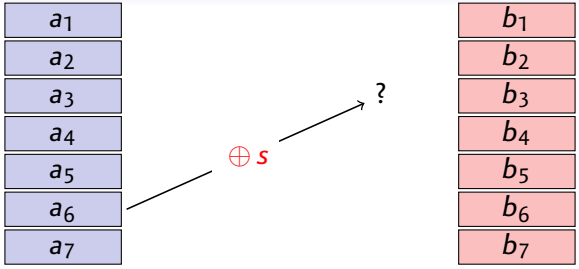
- Make a guess for S, and verify
- Complexity $\tilde{\mathcal{O}}(2^{n/2}\sqrt{|\mathcal{S}|})$ with unbalanced $\mathcal{A}, \mathcal{B}$

Try Guess

for a **in** $\mathcal{A}$ **do**
  if $(s \oplus a) \in \mathcal{B}$ **then**
    return 0
return 1

Searching algorithm

[McGrew, FSE'13]



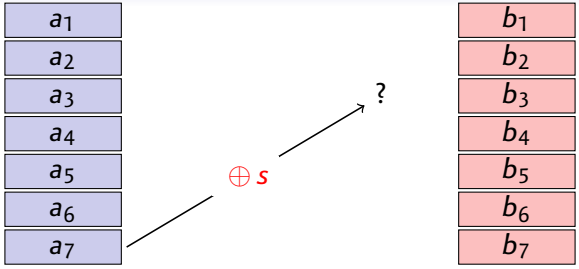
- Make a guess for S, and verify
- Complexity $\tilde{\mathcal{O}}(2^{n/2}\sqrt{|\mathcal{S}|})$ with unbalanced $\mathcal{A}, \mathcal{B}$

Try Guess

for a **in** $\mathcal{A}$ **do**
  if $(s \oplus a) \in \mathcal{B}$ **then**
    return 0
return 1

Searching algorithm

[McGrew, FSE'13]

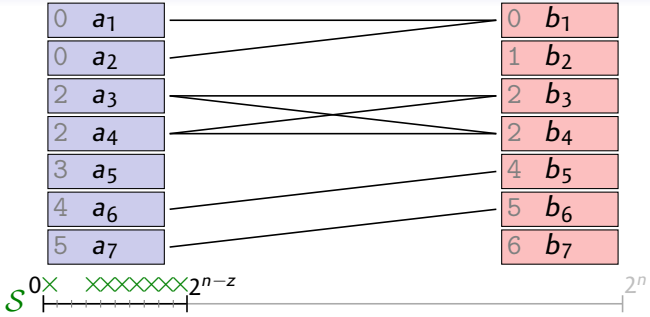


- Make a guess for S, and verify
- Complexity $\tilde{\mathcal{O}}(2^{n/2}\sqrt{|\mathcal{S}|})$ with unbalanced $\mathcal{A}$, $\mathcal{B}$

Try Guess

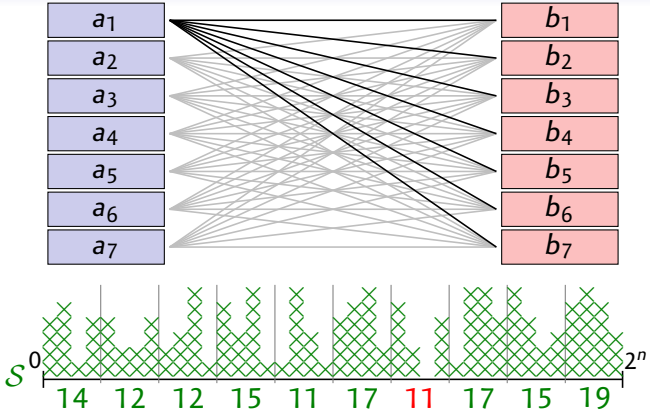
for a **in** $\mathcal{A}$ **do**
  if $(s \oplus a) \in \mathcal{B}$ **then**
    return 0
return 1

Known-prefix sieving



- Assume S starts with z zero bits (more generally, linear subspace)
  - Smaller sieve
- Sort lists, consider a_i's and b_j's with matching prefix
- Complexity: $\tilde{\mathcal{O}}(2^{n/2} + 2^{\dim\langle\mathcal{S}\rangle})$
- Complexity: $\tilde{\mathcal{O}}(2^{n/2})$ when $z \geq n/2$

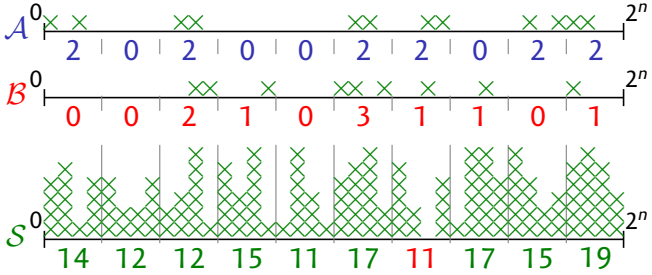
Fast Convolution Sieving



- Instead of computing full sieve, use buckets (*ie.* truncate)
- With enough data, missing difference has smallest bucket with high probability
  - *Eg.* $2^{2n/3}$ queries, sieving with $2^{2n/3}$ buckets of $2^{n/3}$ elements

Computing the sieve

- Count buckets for $\mathcal{A}$ and $\mathcal{B}$
  - $C_{\mathcal{X}}[i] = \left|\{x \in \mathcal{X} \mid T(x) = i\}\right|$



- Convolution can be computed with Fast Walsh-Hadamard transform!
  - Complexity: $\tilde{\mathcal{O}}(2^{2n/3})$ for arbitrary $\mathcal{S}$

Computing the sieve

- Count buckets for $\mathcal{A}$ and $\mathcal{B}$
  - $C_{\mathcal{X}}[i] = \left|\{x \in \mathcal{X} \mid T(x) = i\}\right|$
  - $$\begin{aligned} C_{\mathcal{S}}[i] &= |\{(a,b) \in \mathcal{A} \times \mathcal{B} \mid T(a \oplus b) = i\}| \\ &= \sum_{a \in \mathcal{A}} |\{b \in \mathcal{B} \mid T(a \oplus b) = i\}| \\ &= \sum_{a \in \mathcal{A}} C_{\mathcal{B}}[i \oplus T(a)] \\ &= \sum_{j \in \{0,1\}^{n-t}} C_{\mathcal{A}}[j] C_{\mathcal{B}}[i \oplus j] \end{aligned}$$

- Convolution can be computed with Fast Walsh-Hadamard transform!
  - Complexity: $\tilde{\mathcal{O}}(2^{2n/3})$ for arbitrary $\mathcal{S}$

Computing the sieve

- Count buckets for $\mathcal{A}$ and $\mathcal{B}$
  - $C_{\mathcal{X}}[i] = \left|\{x \in \mathcal{X} \mid T(x) = i\}\right|$
  - $C_{\mathcal{S}}[i] = |\{(a,b) \in \mathcal{A} \times \mathcal{B} \mid T(a \oplus b) = i\}|$

$$= \sum_{a \in \mathcal{A}} |\{b \in \mathcal{B} \mid T(a \oplus b) = i\}|$$

$$= \sum_{a \in \mathcal{A}} C_{\mathcal{B}}[i \oplus T(a)]$$

$$= \sum_{j \in \{0,1\}^{n-t}} C_{\mathcal{A}}[j] C_{\mathcal{B}}[i \oplus j]$$

- Convolution can be computed with Fast Walsh-Hadamard transform!
  - Complexity: $\tilde{\mathcal{O}}(2^{2n/3})$ for arbitrary $\mathcal{S}$

Missing difference problem algorithms

Algorithms for the missing difference problem

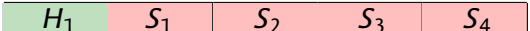
Sieving	Complexity $\tilde{\mathcal{O}}(2^n)$	[McGrew]
Searching	Complexity $\tilde{\mathcal{O}}(2^{n/2}\sqrt{\lvert\mathcal{S}\rvert})$	[McGrew]
Known-prefix sieving	Complexity $\tilde{\mathcal{O}}(2^{n/2} + 2^{\dim\langle\mathcal{S}\rangle})$	
Fast convolution sieving	Complexity $\tilde{\mathcal{O}}(2^{2n/3})$	

- Improved algorithm if $\mathcal{S}$ is a linear subspace
  - Plaintext recovery with CPSS queries
- Improved algorithm with more data for arbitrary $\mathcal{S}$
  - Attacks against Wegman-Carter-Shoup MACs

Application to CTR (CPSS queries)

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

Queries Q_1 with half-block header



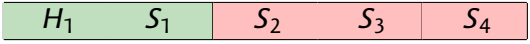
Queries Q_2 with full-block header

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. ...

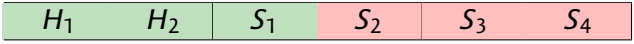
Application to CTR (CPSS queries)

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

Queries Q_1 with half-block header



Queries Q_2 with full-block header

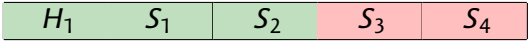


1. Recover S_1 using the first block of each query:
   $\mathcal{A} = \{\mathcal{E}(H_1 \,\|\, H_2)\}, \mathcal{B} = \{\mathcal{E}(H_1 \,\|\, S_1)\}.$ $\rightarrow$ Missing difference: $0 \,\|\, (S_1 \oplus H_2)$.
2. When S_1 is known, recover S_2, with the first and second blocks of Q_2 queries:
  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



Queries Q_2 with full-block header

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. ...

CBC vs. CTR

CBC mode

- Collisions reveals
  xor of two plaintext blocks

CTR mode

- Distinguishing attack:
  Key stream doesn't collide

Message recovery attacks

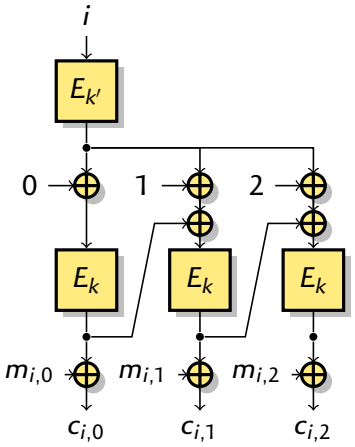
- Both modes have message recovery attacks with birthday complexity
  - Assuming some control over the plaintext, and repeated secret

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.

Use of CTR in practice

- CTR is widely used as the encryption component of GCM
  - 90% of Firefox HTTPS traffic uses AES-GCM
- SSHv2 supports 3DES-CTR
  (one of the recommended ciphers)
  but supported by only 9% of servers
- 3G telephony uses Kasumi (64-bit blocks)
  with the f8 mode
  - First block of keystream does not repeat
  - Nice target, but need some hardware
    to implement the attack...



Wegman-Carter MACs

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

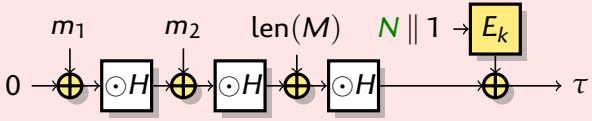
$$\mathrm{WC}(N, M) = H_{k_1}(M) \oplus F_{k_2}(N).$$

$$\mathbf{Adv}^{\mathrm{MAC}}_{\mathrm{WC}[H,F]} \leq \mathbf{Adv}^{\mathrm{PRF}}_{F} + \varepsilon + 2^{-n}$$

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

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

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



Key recovery as a missing difference problem

- Fix two messages $M \neq M'$, capture MACs
  - $a_i = \mathrm{MAC}(i, M) = H(M) \oplus E(i)$
  - $b_j = \mathrm{MAC}(j, M') = H(M') \oplus E(j)$
  - $a_i \oplus b_j \neq H(M) \oplus H(M')$
- For polynomial hashing, easy to recover universal hash key from $H(M) \oplus H(M')$
- Sieving algorithm recovers $H(M) \oplus H(M')$ with $\mathcal{O}(2^{n/2})$ queries and $\mathcal{O}(2^n)$ computations
  - Independently done in another Eurocrypt paper!

Optimal Forgeries Against Polynomial-Based MACs and GCM
Atul Luykx, Bart Preneel [Eurocrypt '18]

- Fast convolution sieving recovers $H(M) \oplus H(M')$ with $\tilde{\mathcal{O}}(2^{2n/3})$ queries and computations
  - First universal forgery attack with less than 2^n operation

Key recovery as a missing difference problem

- Fix two messages $M \neq M'$, capture MACs
  - $a_i = \mathrm{MAC}(i, M) = H(M) \oplus E(i)$
  - $b_j = \mathrm{MAC}(j, M') = H(M') \oplus E(j)$
  - $a_i \oplus b_j \neq H(M) \oplus H(M')$
- For polynomial hashing, easy to recover universal hash key from $H(M) \oplus H(M')$
- Sieving algorithm recovers $H(M) \oplus H(M')$ with $\mathcal{O}(2^{n/2})$ queries and $\mathcal{O}(2^n)$ computations
  - Independently done in another Eurocrypt paper!

  Optimal Forgeries Against Polynomial-Based MACs and GCM
  Atul Luykx, Bart Preneel [Eurocrypt '18]
- Fast convolution sieving recovers $H(M) \oplus H(M')$ with $\tilde{\mathcal{O}}(2^{2n/3})$ queries and computations
  - First universal forgery attack with less than 2^n operation

Key recovery as a missing difference problem

- Fix two messages $M \neq M'$, capture MACs
  - $a_i = \mathrm{MAC}(i, M) = H(M) \oplus E(i)$
  - $b_j = \mathrm{MAC}(j, M') = H(M') \oplus E(j)$
  - $a_i \oplus b_j \neq H(M) \oplus H(M')$
- For polynomial hashing, easy to recover universal hash key from $H(M) \oplus H(M')$

- Sieving algorithm recovers $H(M) \oplus H(M')$ with $\mathcal{O}(2^{n/2})$ queries and $\mathcal{O}(2^n)$ computations
  - Independently done in another Eurocrypt paper!

Optimal Forgeries Against Polynomial-Based MACs and GCM
Atul Luykx, Bart Preneel [Eurocrypt '18]

- Fast convolution sieving recovers $H(M) \oplus H(M')$ with $\tilde{\mathcal{O}}(2^{2n/3})$ queries and computations
  - First universal forgery attack with less than 2^n operation

Conclusion

1. Algorithmic improvements to the missing difference problem
2. Plaintext recovery attack against CTR with birthday complexity
3. Hash key recovery against GMAC, Poly1305-AES with complexity $\tilde{\mathcal{O}}(2^{2n/3})$

- 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
- Data limit matters
  - Adversary can make you generate data
  - NIST requires security up to 2^{50} for lightweight crypto
    - 64-bit block ciphers with classical modes insecure
  - Do we need more than 2^{64} data for conventional crypto?

Conclusion

1. Algorithmic improvements to the missing difference problem
2. Plaintext recovery attack against CTR with birthday complexity
3. Hash key recovery against GMAC, Poly1305-AES with complexity $\tilde{\mathcal{O}}(2^{2n/3})$

- 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
- Data limit matters
  - Adversary can make you generate data
  - NIST requires security up to 2^{50} for lightweight crypto
    - 64-bit block ciphers with classical modes insecure
  - Do we need more than 2^{64} data for conventional crypto?

Security Beyond the Birthday Bound

- Security loss mostly dues to PRF/PRP switching lemma
- 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]
  - CTR with XoP and Wegman-Carter with XoP have 2^n security
- More advanced constructions
  - CENC for encryption [Iwata, FSE'06] [Iwata, Mennink, Vizar '16]
  - EWCDM for authentication [Cogliati & Seurin, Crypto'16]
- Other options
  - Larger state: sponge, Chacha-Poly, ...
  - Tweakable block-cipher, ...