TCC 2026

On the Additive Sensitivity of LZ77 Under Consecutive Edits

Jeremiah Blocki[1] **Seunghoon Lee**[2]

[1]Purdue University [2]**University of Waterloo**

PURDUE UNIVERSITY

UNIVERSITY OF WATERLOO | FACULTY OF MATHEMATICS Department of Combinatorics and Optimization

Compression leaks through length

```
Patient: Jane Doe
Test: SARS-CoV-2 RNA (PCR)
Result: NEGATIVE
Reference: NEGATIVE
```

82 B

same length

```
Patient: Jane Doe
Test: SARS-CoV-2 RNA (PCR)
Result: POSITIVE
Reference: NEGATIVE
```

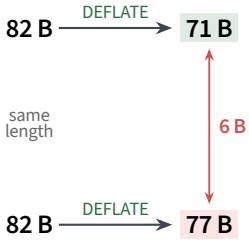
82 B

Compression leaks through length

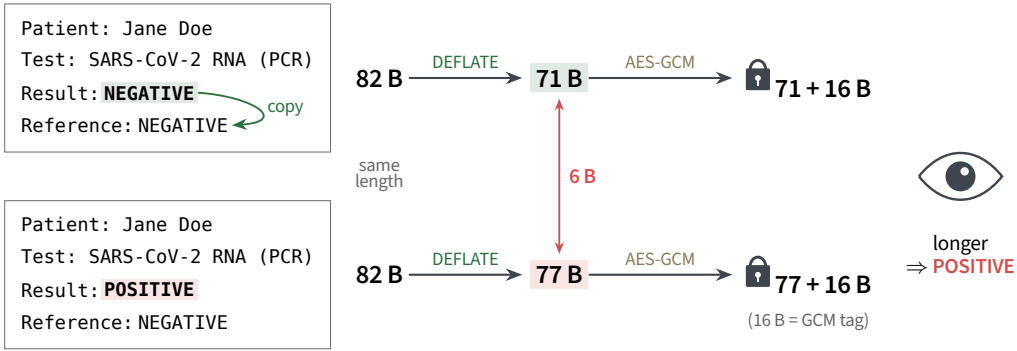
```
Patient: Jane Doe
Test: SARS-CoV-2 RNA (PCR)
Result: NEGATIVE
Reference: NEGATIVE
```

copy

```
Patient: Jane Doe
Test: SARS-CoV-2 RNA (PCR)
Result: POSITIVE
Reference: NEGATIVE
```



Compression leaks through length



Encryption hides the **content**, not the **length** — and compression makes length **depend on content**.

Compression-length leakage has been exploited in practice: CRIME [RD12], BREACH [GHP13].

LZ77 copies, then appends

Example with $W \geq 5$

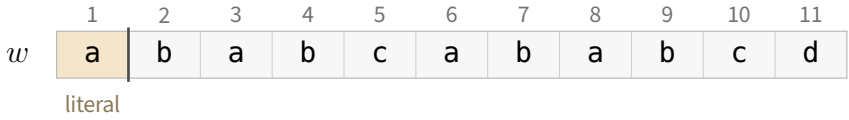
Look back up to W characters, copy the longest match, and append one literal.

	1	2	3	4	5	6	7	8	9	10	11
w	a	b	a	b	c	a	b	a	b	c	d

LZ77 copies, then appends

Example with $W \geq 5$

Look back up to W characters, copy the longest match, and append one literal.

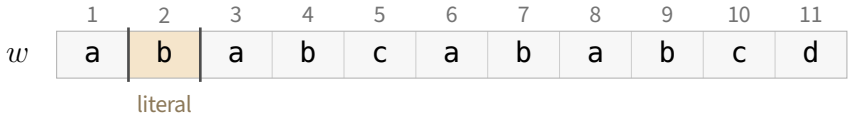


Parse a

LZ77 copies, then appends

Example with $W \geq 5$

Look back up to W characters, copy the longest match, and append one literal.

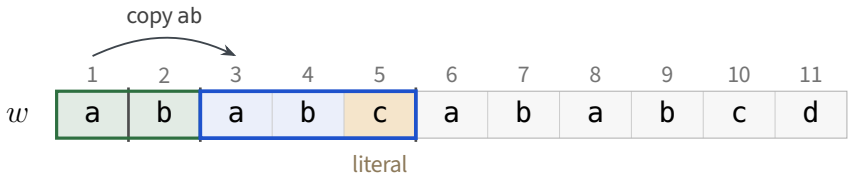


Parse a | b

LZ77 copies, then appends

Example with $W \geq 5$

Look back up to W characters, copy the longest match, and append one literal.

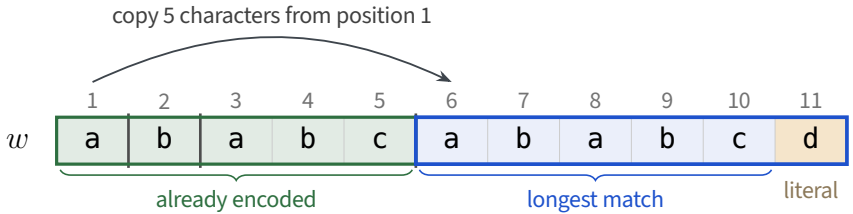


Parse a | b | abc

LZ77 copies, then appends

Example with $W \geq 5$

Look back up to W characters, copy the longest match, and append one literal.



Parse a | b | abc | ababcd

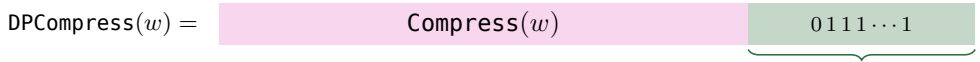
$B_4 = [1, 5, \mathrm{d}]$

source, match length, literal

11 characters $\longrightarrow$ 4 blocks

With a fixed-size block encoding, compressed length is proportional to block count.

Random padding for single-character edits [BLG25]



$M(w) = |\texttt{DPCompress}(w)|$ is (ε, δ)-**DP** if for all neighbors $w \sim w'$ and all sets S of lengths,

$$\Pr[M(w) \in S] \;\leq\; e^{\varepsilon} \Pr[M(w') \in S] + \delta.$$

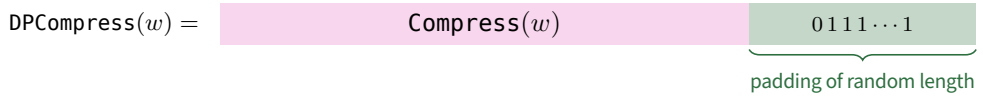
ε controls the multiplicative change in probability; δ allows additive slack. Pure DP: $\delta = 0$.

For fixed ε and $\delta > 0$, padding scale $\propto$ **global sensitivity** GS

$$\text{GS} \;:=\; \max_{w \sim w'} \big|\, |\texttt{Compress}(w)| \;-\; |\texttt{Compress}(w')| \,\big|$$

[BLG25]: LZ77 with window W has $\text{GS} = O(W^{2/3} \log n) = o(n)$ for inputs of length n.

Random padding for single-character edits [BLG25]



$M(w) = |\texttt{DPCompress}(w)|$ is (ε, δ)-**DP** if for all neighbors $w \sim w'$ and all sets S of lengths,

$$\Pr[M(w) \in S] \;\leq\; e^{\varepsilon} \Pr[M(w') \in S] + \delta.$$

ε controls the multiplicative change in probability; δ allows additive slack. Pure DP: $\delta = 0$.

For fixed ε and $\delta > 0$, padding scale $\propto$ **global sensitivity** GS

$$\text{GS} \;:=\; \max_{w \sim w'} \big|\, |\texttt{Compress}(w)| \;-\; |\texttt{Compress}(w')| \,\big|$$

[BLG25]: LZ77 with window W has $\text{GS} = O(W^{2/3} \log n) = o(n)$ for inputs of length n.

Here, **neighbors** $w \sim w'$ differ in **a single character**.

From one character to a whole field

Sensitive data often occupies a contiguous field

- a test result `NEGATIVE` (8 chars)
- a password of fixed length, a PIN, an SSN
- a session cookie, an API token (16–64)
- one person's record in a database file

Neighbors differ in one character [BLG25]



w and w' differ within g **consecutive positions**



From one character to a whole field

Sensitive data often occupies a contiguous field

- a test result `NEGATIVE` (8 chars)
- a password of fixed length, a PIN, an SSN
- a session cookie, an API token (16–64)
- one person's record in a database file

Neighbors differ in one character [BLG25]



w and w' differ within g **consecutive positions**



Privacy guarantee

Group privacy for the existing mechanism [Vad17]

$$(\varepsilon, \delta) \longrightarrow \left(g\varepsilon,\ ge^{g\varepsilon}\delta\right)$$

The guarantee weakens as g grows

Change in compressed length

Triangle inequality over the edits

$$\big|\,|\texttt{Compress}(w)| - |\texttt{Compress}(w')|\,\big| \le g \cdot \mathsf{GS}$$

With this bound, padding scales by g

Both arguments proceed one character at a time, wherever the edits occur.

From one character to a whole field

Sensitive data often occupies a contiguous field

- a test result `NEGATIVE` (8 chars)
- a password of fixed length, a PIN, an SSN
- a session cookie, an API token (16–64)
- one person's record in a database file

Neighbors differ in one character [BLG25]



w and w' differ within g **consecutive positions**



Privacy guarantee

Group privacy for the existing mechanism [Vad17]

$$(\varepsilon, \delta) \longrightarrow \left(g\varepsilon,\ ge^{g\varepsilon}\delta\right)$$

The guarantee weakens as g grows

Change in compressed length

Triangle inequality over the edits

$$\big|\,|\texttt{Compress}(w)| - |\texttt{Compress}(w')|\,\big| \le g \cdot \mathsf{GS}$$

With this bound, padding scales by g

Both arguments proceed one character at a time, wherever the edits occur.

Can we beat $g \cdot \mathsf{GS}$ by analyzing the edited field as a whole?

Sensitivity under contiguous replacements

Main result: upper bound

Two g-neighbors have the same length and agree outside an interval J of g positions.

g-consecutive sensitivity GS^g

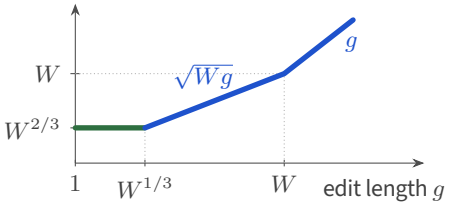
Largest compression length gap between g-neighbors.

$$\mathsf{GS}^g_{\mathsf{LZ77}} = O\Big(\big(W^{2/3} + \sqrt{Wg} + g\big)\log n\Big)$$

LZ77 with window $W \le n$, polynomial-size alphabet.
With or without self-referencing.

Upper-bound scale without $\log n$

Log–log axes



Sensitivity under contiguous replacements

Main result: upper bound

Two g-neighbors have the same length and agree outside an interval J of g positions.

g-consecutive sensitivity GS^g

Largest compression length gap between g-neighbors.

$$\mathsf{GS}^g_{\mathsf{LZ77}} = O\Big(\big(W^{2/3} + \sqrt{Wg} + g\big)\log n\Big)$$

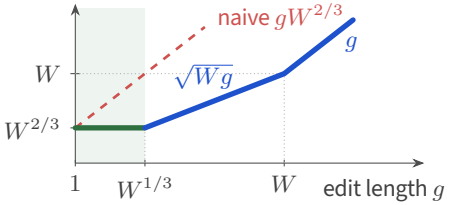
LZ77 with window $W \le n$, polynomial-size alphabet.
With or without self-referencing.

$\sqrt{Wg} \le W^{2/3} \iff g \le W^{1/3}$

$W^{2/3}$ sets the scale up to this threshold.

Upper-bound scale without $\log n$

Log–log axes



Sensitivity under contiguous replacements

Main result: upper bound

Two g-neighbors have the same length and agree outside an interval J of g positions.

g-consecutive sensitivity GS^g

Largest compression length gap between g-neighbors.

$$\mathsf{GS}^g_{\mathsf{LZ77}} = O\Big(\big(W^{2/3} + \sqrt{Wg} + g\big)\log n\Big)$$

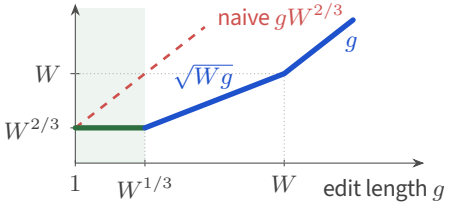
LZ77 with window $W \le n$, polynomial-size alphabet.
With or without self-referencing.

$\sqrt{Wg} \le W^{2/3} \iff g \le W^{1/3}$

$W^{2/3}$ sets the scale up to this threshold.

Upper-bound scale without $\log n$

Log–log axes



Same asymptotic padding scale as for one character for $g \le W^{1/3}$.

Sensitivity under contiguous replacements

Main result: upper bound

Two g-neighbors have the same length and agree outside an interval J of g positions.

g-consecutive sensitivity GS^g

Largest compression length gap between g-neighbors.

$$\mathsf{GS}^g_{\mathsf{LZ77}} = O\Big(\big(W^{2/3} + \sqrt{Wg} + g\big)\log n\Big)$$

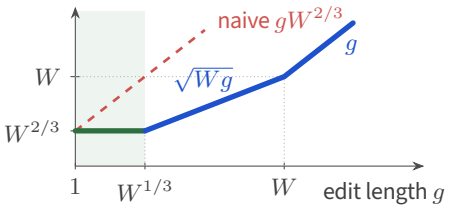
LZ77 with window $W \le n$, polynomial-size alphabet.
With or without self-referencing.

$\sqrt{Wg} \le W^{2/3} \iff g \le W^{1/3}$

$W^{2/3}$ sets the scale up to this threshold.

Upper-bound scale without $\log n$

Log–log axes



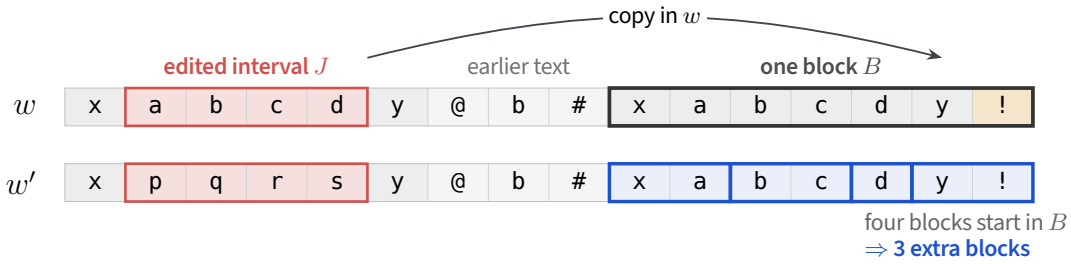
Same asymptotic padding scale as for one character for $g \le W^{1/3}$.

(length difference) = (block count difference) × (bits per block)

Bounding sensitivity by counting blocks

1. Why the compression length can increase

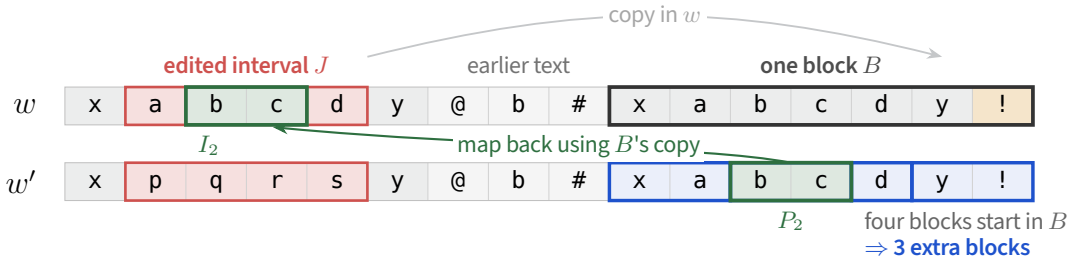
One block B of w can contain several block starts in w'.
Each block has the same bit cost, so extra blocks increase the compressed length.



Bounding sensitivity by counting blocks

1. Why the compression length can increase

One block B of w can contain several block starts in w'.
Each block has the same bit cost, so extra blocks increase the compressed length.



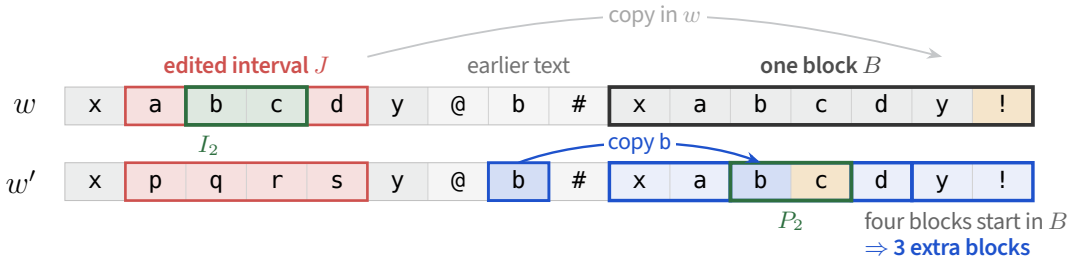
2. Connection to the edited interval

Use B's original copy to map $P_2 = \texttt{bc}$ back to I_2 inside J.
We count the pieces that map entirely into J.

Bounding sensitivity by counting blocks

1. Why the compression length can increase

One block B of w can contain several block starts in w'.
Each block has the same bit cost, so extra blocks increase the compressed length.



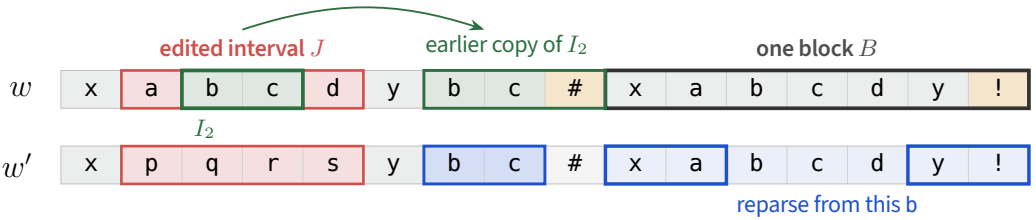
2. Connection to the edited interval

Use B's original copy to map $P_2 = \texttt{bc}$ back to I_2 inside J.
We count the pieces that map entirely into J.

Why a split piece needs a fresh interval

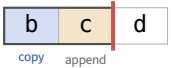
3. Freshness

Suppose an earlier block of w, after J, had copied $I_2 \subseteq J$. That copy would still be available in w'.



Without an earlier bc

The longest match is b.

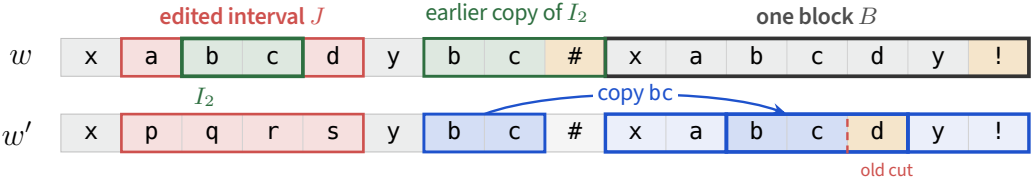


Ends at c.

Why a split piece needs a fresh interval

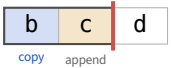
3. Freshness

Suppose an earlier block of w, after J, had copied $I_2 \subseteq J$. That copy would still be available in w'.



Without an earlier bc

The longest match is b.



Ends at c.

With an earlier bc

At the same start in w'.



Reaches at least d.

LZ77 could copy the whole piece and append another character, contradicting the observed cut.

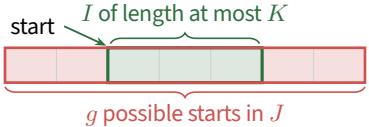
So the same interval of J cannot be counted again.

Two counts give the square root

Split the pieces at length K: fresh subintervals of J bound the short ones, space bounds the long ones.

Short pieces ($\leq K$)

Source intervals in J



At most g starts and K lengths per start.
Each pair is fresh at most once.

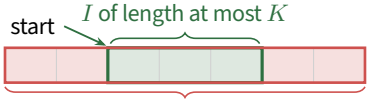
$\leq gK$ short pieces

Two counts give the square root

Split the pieces at length K: fresh subintervals of J bound the short ones, space bounds the long ones.

Short pieces ($\leq K$)

Source intervals in J



At most g starts and K lengths per start.
Each pair is fresh at most once.

$\leq gK$ **short pieces**

Long pieces ($> K$)

Target positions after J



Each piece lies in B's copied part.
B starts at most W positions after J.
The copied part has length at most W.

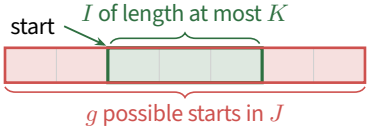
$\leq 2W/K$ **long pieces**

Two counts give the square root

Split the pieces at length K: fresh subintervals of J bound the short ones, space bounds the long ones.

Short pieces ($\leq K$)

Source intervals in J

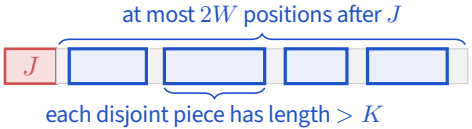


At most g starts and K lengths per start.
Each pair is fresh at most once.

$\leq gK$ **short pieces**

Long pieces ($> K$)

Target positions after J



Each piece lies in B's copied part.
B starts at most W positions after J.
The copied part has length at most W.

$\leq 2W/K$ **long pieces**

$$(\#\text{ counted pieces}) \;\leq\; gK + \frac{2W}{K} \;=\; O\big(\sqrt{Wg}\big) \qquad \text{choose } K = \sqrt{W/g},\; g \leq W.$$

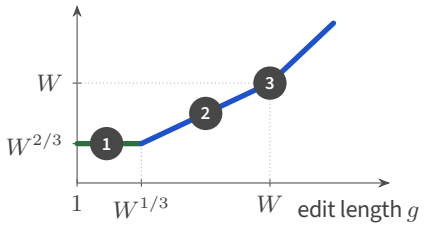
Adding $O(g)$ for starts in J and $O(W^{2/3})$ for source-boundary cases:

$$(\#\text{ extra blocks}) = O\big(W^{2/3} + \sqrt{Wg} + g\big) \;\Rightarrow\; (\text{length difference}) = O\big((W^{2/3} + \sqrt{Wg} + g)\log n\big)$$

Each term is necessary in some regime

Lower bounds for non-self-referencing LZ77

Upper-bound scale



1. $W^{2/3}$ [BLG25], [FSNI25]
   already for one character: $g = 1, W = n$
2. $\sqrt{Wg}$ this work
   $W = n$ and $g \geq W^{1/3}$
3. g this work
   g, $W = \Theta(n)$

Matching lower bounds show that all three terms are necessary.

Large alphabets: matching up to constants. Binary alphabet: up to log factors.

Takeaway

We bound how much LZ77 compressed length can change under g consecutive replacements.

Calibrating padding to this bound gives direct (ε, δ)-DP for g-neighbors.

$$\mathsf{GS}^{g}_{\mathsf{LZ77}} = O\Big(\big(W^{2/3} + \sqrt{W g} + g\big)\log n\Big) \text{ instead of } g \cdot \mathsf{GS}_{\mathsf{LZ77}} = O\big(g\, W^{2/3} \log n\big)$$

For $g \leq W^{1/3}$, the padding has the same asymptotic order as for one edit.

Each term is necessary in some regime, so the bound is nearly tight.

Open questions

- Tight bounds for contiguous insertions and deletions?
- Characterize the sensitivity of other compression schemes, or design new ones with good compression ratios and low sensitivity.

Takeaway

We bound how much LZ77 compressed length can change under g consecutive replacements.

Calibrating padding to this bound gives direct (ε, δ)-DP for g-neighbors.

$$\mathsf{GS}^{g}_{\mathsf{LZ77}} = O\Big(\big(W^{2/3} + \sqrt{W g} + g\big)\log n\Big) \text{ instead of } g \cdot \mathsf{GS}_{\mathsf{LZ77}} = O\big(g\, W^{2/3} \log n\big)$$

For $g \leq W^{1/3}$, the padding has the same asymptotic order as for one edit.

Each term is necessary in some regime, so the bound is nearly tight.

Open questions

- Tight bounds for contiguous insertions and deletions?
- Characterize the sensitivity of other compression schemes, or design new ones with good compression ratios and low sensitivity.

Thank you!

References

[AFI23] T. Akagi, M. Funakoshi, S. Inenaga. Sensitivity of string compressors and repetitiveness measures. *Inf. Comput.* 291, 2023.

[BLG25] J. Blocki, S. Lee, B. S. Yepes-Garcia. Differentially private compression and the sensitivity of LZ77. *TCC 2025*.

[FSNI25] Y. Fujie, H. Shibata, Y. Nakashima, S. Inenaga. Tight additive sensitivity on LZ-style compressors and string attractors. *SPIRE 2025*.

[GHP13] Y. Gluck, N. Harris, A. Prado. BREACH: Reviving the CRIME attack. 2013.

[RD12] J. Rizzo, T. Duong. The CRIME attack. *Ekoparty* 2012.

[Vad17] S. Vadhan. The complexity of differential privacy. *Tutorials on the Foundations of Cryptography*, 2017.