TECHNISCHE UNIVERSITÄT DRESDEN

Scientific Writing in Computer Science

Martin Byrenheid & Paul Walther
Chair of Privacy and Data Security

Outline

- **Learning goal:** Effectively communicate your findings

Outline

- **Learning goal:** Effectively communicate your findings
- **In the following:**
  1. Typical outline of scientific papers
  2. Scientific writing style
  3. Dealing with writer's block

Paper Structure

- Abstract
- Introduction
- Related Work
- Background
- Main part
- Conclusion & Future Work
- Two-column or single column layout

2009 30th IEEE Symposium on Security and Privacy

De-anonymizing Social Networks

Arvind Narayanan and Vitaly Shmatikov
The University of Texas at Austin

Abstract

Operators of online social networks are increasingly sharing potentially sensitive information about users and their relationships with advertisers, application developers, and data-mining researchers. Privacy is typically protected by anonymization, i.e., removing names, addresses, etc.

We present a framework for analyzing privacy and anonymity in social networks and develop a new re-identification algorithm targeting anonymized social-network graphs. To demonstrate its effectiveness on real-world networks, we show that a third of the users who can be verified to have accounts on both Twitter, a popular microblogging service, and Flickr, an online photo-sharing site, can be re-identified in the anonymous Twitter graph with only a 12% error rate.

Our de-anonymization algorithm is based purely on the network topology, does not require creation of a large number of dummy "sybil" nodes, is robust to noise and all existing defenses, and works even when the overlap between the target network and the adversary's auxiliary information is small.

1. Introduction

Social networks have been studied for a century [66] and are a staple of research in disciplines such as epidemiology [8], sociology [73], [28], [11], economics [29], and many others [19], [9], [32]. The recent proliferation of online social networks such as MySpace, Facebook, Twitter, and so on has attracted attention of computer scientists, as well [40].

Even in the few online networks that are completely open, there is a disconnect between users' willingness to share information and their reaction to unintended parties viewing or using this information [13]. Most operators thus provide at least some privacy controls. Many online and virtually all offline networks (*e.g.*, telephone calls, email and instant messages, *etc.*) restrict access to the information about individual members and their relationships.

Network owners often share this information with advertising partners and other third parties. Such sharing is the foundation of the business case for many online social-network operators. Some networks are even published for research purposes. To alleviate privacy concerns, the networks are *anonymized*, *i.e.*, names and demographic information associated with individual nodes are suppressed. Such suppression is often misinterpreted as removal of "personally identifiable information" (PII), even though PII may include much more than names and identifiers. For example, the EU privacy directive defines "personal data" as "any information relating to an identified or identifiable natural person [...]; an identifiable person is one who can be identified, directly or indirectly, in particular by reference to an identification number or to one or more factors specific to his physical, physiological, mental, economic, cultural or social identity" [22].

Anonymity has been unquestioningly interpreted as equivalent to privacy in several high-profile cases of data sharing. After a New York court ruling ordering Google to hand over viewing data of over 100 million YouTube users to Viacom and the subsequent protests from privacy advocates, a revised agreement was struck under which Google would anonymize the data before handing it over [71]. The CEO of NebuAd, a U.S. company that offers targeted advertising based on browsing histories gathered from ISPs, dismissed privacy concerns by saying that "We don't have any raw data on the identifiable individual. Everything is anonymous" [15]. Phorm, a U.K. company with a similar business model, aims to collect the data on Web-surfing habits of 70% of British broadband users; the only privacy protection is that user identities are mapped to random identifiers [69]. In social networks, too, user anonymity has been used as the answer to all privacy concerns (see Section 2).

Our contributions. This is the first paper to demonstrate feasibility of large-scale, passive de-anonymization of real-world social networks.

First, we survey the current state of data sharing in social networks, the intended purpose of each type of sharing, the resulting privacy risks, and the wide availability of auxiliary information which can aid the attacker in de-anonymization.

Second, we formally define privacy in social networks and relate it to node anonymity. We identify several categories of attacks, differentiated by attackers' resources and auxiliary information. We also give a methodology for measuring the extent of privacy breaches in social networks, which is an interesting problem in its own right.

Third, we develop a generic re-identification algorithm for anonymized social networks. The algorithm uses only the network structure, does not make any *a priori* assumptions about membership overlap between multiple networks, and defeats all known defenses.

1081-6011/09 $25.00 © 2009 IEEE
DOI 10.1109/SP.2009.22

Abstract

- What is the problem you are addressing?
- Why is it an important problem?
- What is your approach?
- Which evidence do you have?

Abstract

- What is the problem you are addressing?
- Why is it an important problem?
- What is your approach?
- Which evidence do you have?

Kamvar et al. “The Eigentrust Algorithm for Reputation Management in P2P Networks”, WWW 2003:

ABSTRACT

Peer-to-peer file-sharing networks are currently receiving much attention as a means of sharing and distributing information. However, as recent experience shows, the anonymous, open nature of these networks offers an almost ideal environment for the spread of self-replicating inauthentic files.

We describe an algorithm to decrease the number of downloads of inauthentic files in a peer-to-peer file-sharing network that assigns each peer a unique global trust value, based on the peer’s history of uploads. We present a distributed and secure method to compute global trust values, based on Power iteration. By having peers use these global trust values to choose the peers from whom they download, the network effectively identifies malicious peers and isolates them from the network.

In simulations, this reputation system, called EigenTrust, has been shown to significantly decrease the number of inauthentic files on the network, even under a variety of conditions where malicious peers cooperate in an attempt to deliberately subvert the system.

Introduction

- What is the more general problem you are addressing?
- Which part of the problem do you address?
- Why is it an important part?
- (If existing:) Why are existing solutions insufficient?
- What is your approach?
- Which evidence do you have to proof its superiority?

Further sections

- Background
  - Contains necessary information for main part
- Related Work
  - Shortly explain existing works addressing your problem
  - Outline gap that existing works leave open
- Conclusion & Future Work
  - Briefly summarize your contributions and results
  - Outline open questions and problems

Writing style

- Start with notes for yourself, then incrementally adapt the text for your audience
- What is your audience?
  - Necessary background knowledge
  - Expected volume of information

Writing style

Guiding questions for outline of main part:

- What messages / insights do I want the reader to learn?
- What makes each insight plausible?
- How do all the insights fit into the bigger picture?
- For background: which prior knowledge is needed for my audience to understand the insights?

Writing style

- Each paragraph corresponds to one message / insight
- Allows reader to take breaks
- Use figures for illustration

Writing style

- Each paragraph corresponds to one message / insight
- Allows reader to take breaks
- Use figures for illustration

The distance metric then defines the distance between two logical coordinates and thus also serves as measure of distance between nodes. An assignment of logical coordinates to nodes is called a *greedy embedding* if greedy routing, i.e., forwarding the message to the neighbor whose logical coordinate has the lowest distance to the coordinate of the target, is guaranteed to succeed. In this work, we focus on vector-based coordinates of varying length, i.e., $\mathbf{ID} = S^*$ for some set S. The assignment is executed in a fully distributed manner as follows: First, a single node r out of all nodes in the network is selected and receives a coordinate. Starting from r, a spanning tree of the network rooted at r is constructed. Children receive a coordinate that is the parent coordinate and one additional element. In other words, whenever a node u becomes the child of another node v with logical coordinate $(c_1, c_2, ..c_k)$, v's coordinate is of the form $(c_1, c_2, .., c_k, c_u)$ for some $c_u \in S$. Thus, the vector assigned to u encodes a path from u to the root node [8], [9], [20]. Since a rooted spanning tree is a connected subgraph over the entire network, all nodes can reach each other by routing over the tree edges. Thus the distance between two logical coordinates C_1 and C_2 is given by means of the *tree distance* $h_{TD}(C_1, C_2) = |C_1| + |C_2| - 2 \cdot CPL(C_1, C_2)$, where $CPL(C_a, C_b)$ denotes the length of the common prefix of C_a and C_b. However, the routing of messages is not limited to tree edges only. When routing a message to a coordinate C_t, nodes do not only consider the distance of the logical coordinates of their parent and children to C_t but those of *all* their neighbors in the network. In the following, we call non-tree edges *shortcuts*, as these links can be used to reduce the number of hops needed to reach C_t. In this work, we consider an adversary that aims to perform a large-scale denial of service attack against the overlay network. For overlay networks such as Freenet or GNUnet, the adversary might be a malicious actor that aims to perform censorship. In Lightning, the attacker might want to block payments such that parties make use of other payment methods with higher fees. We consider an *internal* attack, where the adversary controls a subset of the nodes in the overlay. In the following, we call nodes under control of the adversary *malicious nodes* and the remaining nodes are called *benign nodes*. As the initial setup of connections in topology-restricted overlay networks requires prior social engineering, which we assume to be costly to perform on a large-scale, the adversary can only establish a bounded number of connections between malicious and benign nodes. In the following, we call connections between malicious and benign nodes *attack edges*. The malicious nodes may deviate arbitrarily from the correct behavior, e.g. by dropping and delaying messages or spreading misinformation. In particular, we consider the scenario that the malicious nodes are able to undermine the election of the root node, thus establishing a malicious node as root. Since we do not assume a centralized admission control, the adversary is furthermore able to simulate additional, arbitrarily interconnected nodes in the network, as illustrated by Figure ??. In the following, we call the network of adversary-controlled nodes together with the simulated nodes the *adversarial region* of the overlay. The network of non-adversarial nodes is called the *benign region* of the overlay. However, we assume that the adversary does not have any a priori knowledge about the total number of benign nodes and their connections. Rather, he is initially only aware of those benign nodes that are connected to malicious nodes. He is furthermore non-adaptive in the sense that he establishes his connections initially and does not add or remove connections later.

IV. Attacks and countermeasures

Given that our adversary can only establish a bounded number of attack edges to benign nodes, the adversary uses these edges to perform active attacks in order to maximize the disruption of communication. In the scenario that a malicious node has been elected as root, the attacker can perform different attacks by varying the length and elements of the logical coordinates sent via the attack edges as well as the timing of these messages. Given these attack vectors, the adversary may perform the following attacks: Coordinate duplication: Malicious nodes propose the same logical coordinate to multiple benign neighbors. Simulate high diameter: Malicious nodes announce extremely long coordinates to their benign neighbors. Simulate high dynamics: Malicious nodes simulate extreme dynamics in the adversarial region by repeatedly announcing different logical coordinates to their benign neighbors. Simulate root fault: Malicious nodes never announce any logical coordinates to their benign neighbors, thus pretending to have lost connectivity to the root node after the election. The first attack causes routing to fail, as the assignment of logical coordinates is not unique anymore, such that benign nodes forward messages to the wrong nodes. However, we merely include this attack for completeness, as it has already been addressed by Roos et al. [20] by having child nodes obtain their coordinate by appending a random number to the coordinate of their parent. The second attack does not cause routing to fail, but instead introduces extremely high bandwidth overhead due to excessively long addresses, which significantly lowers throughput. In the presence of a malicious root node, routing between different subtrees depends the usage of shortcut links. By sending coordinates with different lengths over each attack edge, the third attack causes the benign nodes to frequently change their parents and consequently their logical addresses. As a result, the target coordinate of messages that are in transit become outdated and are routed towards the malicious root node, as benign nodes are unable to detect shortcuts. In case of the fourth attack, benign nodes will not obtain any logical addresses, thus making routing of messages impossible. One intuitive countermeasure that limits the damage caused by the aforementioned attacks is to periodically start a new election after a fixed amount of time, starting from a common fixed date. However, while a shorter election period reduces the timespan of the attacks, it also inherently causes higher overhead in the absence of attacks. It is thus desirable to design countermeasures that limit the damage caused by malicious nodes while one of them still acts as root node.

Writing style

- Each paragraph corresponds to one message / insight
- Allows reader to take breaks
- Use figures for illustration

The distance metric then defines the distance between two logical coordinates and thus also serves as measure of distance between nodes. An assignment of logical coordinates to nodes is called a *greedy embedding* if greedy routing, i.e., forwarding the message to the neighbor whose logical coordinate has the lowest distance to the coordinate of the target, is guaranteed to succeed.

In this work, we focus on vector-based coordinates of varying length, i.e., $\mathbf{ID} = S^*$ for some set S. The assignment is executed in a fully distributed manner as follows: First, a single node r out of all nodes in the network is selected and receives a coordinate. Starting from r, a spanning tree of the network rooted at r is constructed. Children receive a coordinate that is the parent coordinate and one additional element. In other words, whenever a node u becomes the child of another node v with logical coordinate $(c_1, c_2, ..., c_k)$, u's coordinate is of the form $(c_1, c_2, ..., c_k, c_u)$ for some $c_u \in S$. Thus, the vector assigned to u encodes a path from u to the root node [8], [9], [20].

Since a rooted spanning tree is a connected subgraph over the entire network, all nodes can reach each other by routing over the tree edges. Thus the distance between two logical coordinates C_1 and C_2 is given by means of the *tree distance*

$$\delta_{TD}(C_1, C_2) = |C_1| + |C_2| - 2 \cdot CPL(C_1, C_2) \quad (1)$$

, where $CPL(C_x, C_y)$ denotes the length of the common prefix of C_x and C_y.

However, the routing of messages is not limited to tree edges only. When routing a message to a coordinate C_t, nodes do not only consider the distance of the logical coordinates of their parent and children to C_t but those of all their neighbors in the network. In the following, we call non-tree edges *shortcuts*, as these links can be used to reduce the number of hops needed to reach C_t.

C. Adversary model

In this work, we consider an adversary that aims to perform a large-scale denial of service attack against the overlay network. For overlay networks such as Freenet or GNUnet, the adversary might be a malicious actor that aims to perform censorship. In Lightning, the attacker might want to block payments such that parties make use of other payment methods with higher fees.

We consider an *internal* attack, where the adversary controls a subset of the nodes in the overlay. In the following, we call nodes under control of the adversary *malicious nodes* and the remaining nodes are called *benign nodes*.

As the initial setup of connections in topology-restricted overlay networks requires prior social engineering, which we assume to be costly to perform on a large-scale, the adversary can only establish a bounded number of connections between malicious and benign nodes. In the following, we call connections between malicious and benign nodes *attack edges*.

The malicious nodes may deviate arbitrarily from the correct behavior, e.g. by dropping and delaying messages or spreading misinformation. In particular, we consider the scenario that the malicious nodes are able to undermine the election of the root node, thus establishing a malicious node as root.

Since we do not assume a centralized admission control, the adversary is furthermore able to simulate additional, arbitrarily interconnected nodes in the network, as illustrated by Figure 1. In the following, we call the network of adversary-controlled nodes together with the simulated nodes the *adversarial region* of the overlay. The network of non-adversarial nodes is called the *benign region* of the overlay.

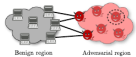


Fig. 1: The adversary is able to introduce fake nodes (indicated by transparency) with arbitrary interconnections. Thus, the root node, marked by a dashed line, may also be a fake node.

However, we assume that the adversary does not have any a priori knowledge about the total number of benign nodes and their connections. Rather, he is initially only aware of those benign nodes that are connected to malicious nodes. He is furthermore non-adaptive in the sense that he establishes his connections initially and does not add or remove connections later.

IV. Attacks and countermeasures

Given that our adversary can only establish a bounded number of attack edges to benign nodes, the adversary uses these edges to perform active attacks in order to maximize the disruption of communication. In the scenario that a malicious node has been elected as root, as shown in Figure 1, the attacker can perform different attacks by varying the length and elements of the logical coordinates sent via the attack edges as well as the timing of these messages.

Given these attack vectors, the adversary may perform the following attacks:

1) **Coordinate duplication:** Malicious nodes propose the same logical coordinate to multiple benign neighbors.
2) **Simulate high diameter:** Malicious nodes announce extremely long coordinates to their benign neighbors.
3) **Simulate high dynamics:** Malicious nodes simulate extreme dynamics in the adversarial region by repeatedly announcing different logical coordinates to their benign neighbors.
4) **Simulate root fault:** Malicious nodes never announce any logical coordinates to their benign neighbors, thus pretending to have lost connectivity to the root node after the election.

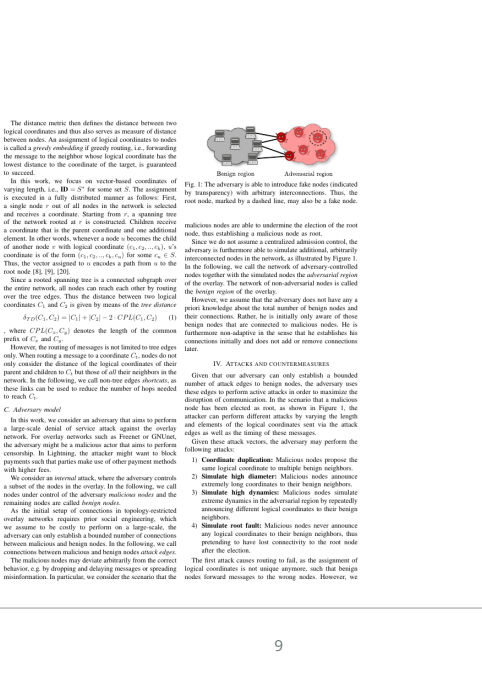
The first attack causes routing to fail, as the assignment of logical coordinates is not unique anymore, such that benign nodes forward messages to the wrong nodes. However, we

Writing style

- Each paragraph corresponds to one message / insight
- Allows reader to take breaks
- Use figures for illustration

Guiding principles from Dreyer:

- **Flow:** It should be clear how each sentence and paragraph relates to the adjacent ones.
- **Coherence:** It should be clear how each sentence and paragraph relates to the big picture.



The distance metric then defines the distance between two logical coordinates and thus also serves as measure of distance between nodes. An assignment of logical coordinates to nodes is called a *greedy embedding* if greedy routing, i.e., forwarding the message to the neighbor whose logical coordinate has the lowest distance to the coordinate of the target, is guaranteed to succeed.

In this work, we focus on vector-based coordinates of varying length, i.e., $\mathbf{ID} = S^*$ for some set S. The assignment is executed in a fully distributed manner as follows: First, a single node r out of all nodes in the network is selected and receives a coordinate. Starting from r, a spanning tree of the network rooted at r is constructed. Children receive a coordinate that is the parent coordinate and one additional element. In other words, whenever a node u becomes the child of another node v with logical coordinate $(c_1, c_2, ..., c_k)$, u's coordinate is of the form $(c_1, c_2, ..., c_k, c_u)$ for some $c_u \in S$. Thus, the vector assigned to u encodes a path from u to the root node [8], [9], [20].

Since a rooted spanning tree is a connected subgraph over the entire network, all nodes can reach each other by routing over the tree edges. Thus the distance between two logical coordinates C_1 and C_2 is given by means of the *tree distance*

$$\delta_{TD}(C_1, C_2) = |C_1| + |C_2| - 2 \cdot CPL(C_1, C_2) \quad (1)$$

, where $CPL(C_x, C_y)$ denotes the length of the common prefix of C_x and C_y.

However, the routing of messages is not limited to tree edges only. When routing a message to a coordinate C_t, nodes do not only consider the distance of the logical coordinates of their parent and children to C_t but those of *all* their neighbors in the network. In the following, we call non-tree edges *shortcuts*, as these links can be used to reduce the number of hops needed to reach C_t.

C. Adversary model

In this work, we consider an adversary that aims to perform a large-scale denial of service attack against the overlay network. For overlay networks such as Freenet or GNUnet, the adversary might be a malicious actor that aims to perform censorship. In Lightning, the attacker might want to block payments such that parties make use of other payment methods with higher fees.

We consider an *internal* attack, where the adversary controls a subset of the nodes in the network. In the following, we call nodes under control of the adversary *malicious nodes* and the remaining nodes are called *benign nodes*.

As the initial setup of connections in topology-restricted overlay networks requires prior social engineering, which we assume to be costly to perform on a large-scale, the adversary can only establish a bounded number of connections between malicious and benign nodes. In the following, we call connections between malicious and benign nodes *attack edges*.

The malicious nodes may deviate arbitrarily from the correct behavior, e.g. by dropping and delaying messages or spreading misinformation. In particular, we consider the scenario that the malicious nodes are able to undermine the election of the root node, thus establishing a malicious node as root.

Since we do not assume a centralized admission control, the adversary is furthermore able to simulate additional, arbitrarily interconnected nodes in the network, as illustrated by Figure 1. In the following, we call the network of adversary-controlled nodes together with the simulated nodes the *adversarial region* of the overlay. The network of non-adversarial nodes is called the *benign region* of the overlay.

Benign region Adversarial region

Fig. 1: The adversary is able to introduce fake nodes (indicated by transparency) with arbitrary interconnections. Thus, the root node, marked by a dashed line, may also be a fake node.

However, we assume that the adversary does not have any a priori knowledge about the total number of benign nodes and their connections. Rather, he is initially only aware of those benign nodes that are connected to malicious nodes. He is furthermore non-adaptive in the sense that he establishes his connections initially and does not add or remove connections later.

IV. Attacks and countermeasures

Given that our adversary can only establish a bounded number of attack edges to benign nodes, the adversary uses these edges to perform active attacks in order to maximize the disruption of communication. In the scenario that a malicious node has been elected as root, as shown in Figure 1, the attacker can perform different attacks by varying the length and elements of the logical coordinates sent via the attack edges as well as the timing of these messages.

Given these attack vectors, the adversary may perform the following attacks:

1) **Coordinate duplication:** Malicious nodes propose the same logical coordinate to multiple benign neighbors.
2) **Simulate high diameter:** Malicious nodes announce extremely long coordinates to their benign neighbors.
3) **Simulate high dynamics:** Malicious nodes simulate extreme dynamics in the adversarial region by repeatedly announcing different logical coordinates to their benign neighbors.
4) **Simulate root fault:** Malicious nodes never announce any logical coordinates to their benign neighbors, thus pretending to have lost connectivity to the root node after the election.

The first attack causes routing to fail, as the assignment of logical coordinates is not unique anymore, such that benign nodes forward messages to the wrong nodes. However, we

Writing style

- Each paragraph corresponds to one message / insight
- Allows reader to take breaks
- Use figures for illustration

Guiding principles from Dreyer:

- **Flow:** It should be clear how each sentence and paragraph relates to the adjacent ones.
- **Coherence:** It should be clear how each sentence and paragraph relates to the big picture.

One of the first approaches to obtain a snapshot of the Internet was by means of sending IP packets with varying initial values in their Time-To-Live (TTL) field [9], [13], [15], [23]. Whenever the TTL of an IP packet reaches zero during transit, many Internet routers send a notification towards the sender of the message. As the notification contains the IP address of the reporting router, paths between different endpoints can be recovered by sending packets with increasing initial TTL values between them while recording the received notification messages. Since state-of-the-art embedding-based routing algorithms for F2F overlays do not employ a notification mechanism for dropped packets, the aforementioned approach is not applicable.

Works from the area of *network tomography* infer the topology between multiple nodes based on end-to-end probe measurements of network characteristics, such as message loss or delay [7], [17], [18], [20]. If there is a high correlation between two nodes u and v when probes are sent by the same node n, then it is assumed that the path from n to u overlaps with the path from n to v and thus, there must be a common node w on both of the paths.

However, tomography can detect if paths are likely to overlap but cannot reveal the number of overlapping nodes or the actual length of the paths. Thus, the inferred topology may contain fewer nodes than there actually are. To overcome this limitation, network tomography approaches have been extended to leverage notification messages [20] or packets with a limited hop number [18]. As mentioned before, approaches based on notification about dropped messages are not applicable to current F2F overlays and since greedy embeddings do not suffer from routing loops, limiting the maximum number of hops is unnecessary.

Writing style

For more examples:

Derek Dreyer - How to Write Papers So People Can Read Them

`https://www.youtube.com/watch?v=L_6xoMjFr70`

Analytical Writing

`https://www.youtube.com/watch?v=lKavD1BTN1A`

Writing style

- Focus on clarity when writing
- Avoid too much jargon
- Use active instead of passive voice
- Do not making strong statements without evidence
- When using figures, refer to them in the text
- Use a spell checker

Writing style

Lessons by Baltimore Writing Center, University of Maryland:

Clear and Effective Prose
`https://www.youtube.com/watch?v=5ccqwEHeTgo`

Active versus passive voice
`https://www.youtube.com/watch?v=ksioYG5EUXM`

Jargon & "Jargonitis"
`https://www.youtube.com/watch?v=Uygqma-AwKI`

Using “real” verbs
`https://www.youtube.com/watch?v=qzjZiRCW96w`

Citation

- Grammatical correctness should not depend on presence of citation
  - **Wrong:** "The reputation system in [9] can significantly decrease the number of inauthentic files.”
  - **Better:** "The reputation system proposed by Kamvar et al. [9] can significantly decrease the number of inauthentic files.”
  - **Better:** "Reputation systems can significantly decrease the number of inauthentic files [9].”

Citation

- Try to avoid placement of citations at ambiguous places[1]
  - **Wrong:** The original algorithm has asymptotic cost $O(n^2)$ but low memory usage, so it is not entirely superseded by Ahlberg's approach, which although of cost $O(n \log n)$ requires a large in-memory array [1,2].
  - **Better:** The original algorithm has asymptotic cost $O(n^2)$ but low memory usage [1], so it is not entirely superseded by Ahlberg's approach [2], which although of cost $O(n \log n)$ requires a large in-memory array.

[1] Example taken from Zobel, Justin. Writing for computer science. 3rd ed., Springer, 2014.

Citation & Plagiarism

- Even if you find an explanation or introduction to be very well-written, do not copying text passages from other works
  - → Violates good scientific practice
  - → Is at least an attempted deception, if not even illegal
  - → Raises suspicion that you did not actually think for yourself

- Further reading:
  `www.ou.edu/content/dam/integrity/docs/nine_things_you_should_know.pdf`

Writer’s block

– See writing as work like any other

Writer's block

- See writing as work like any other
- Dealing with temporary lack of motivation:
  - Plan your writing and set goals
  - Communicate your goals to others
  - Set specific times to write
  - Establish rituals to get you into writing mode
  - Start to write freely and incrementally improve your text

Summary

- Typical outline of scientific papers
  - Abstract, Introduction, Background,..
- Writing style
  - Adapt writing to your audience
  - Focus on clarity and ease of understanding
- Writer's block
  - Writing is hard work like any other
  - Develop methods to deal with lack of motivation