Security in Wiki-Style Authoring Systems

Christian Damsgaard Jensen

Department of Informatics and Mathematical Modelling
Technical University of Denmark
Christian.Jensen@imm.dtu.dk

Summary. During the past decade, online collaboration has grown from a practice primarily associated with the workplace to a social phenomenon, where ordinary people share information about their life, hobbies, interests, politics etc. In particular, social software, such as open collaborative authoring systems like wikis, has become increasingly popular. This is probably best illustrated through the immense popularity of the Wikipedia, which is a free encyclopedia collaboratively edited by thousands of Internet users with a minimum of administration.

As more and more people come to rely on the information stored in open collaborative authoring systems, security is becoming an important concern for such systems. Inaccuracies in the Wikipedia have been rumoured to cause students to fail courses, innocent people have been associated with the murder of John F. Kennedy, etc. Improving the correctness, completeness and integrity of information in collaboratively authored documents is therefore of vital importance to the continued success of such systems.

It has previously been observed that integrity is the most important security property in open collaborative authoring systems. In this paper we propose a general security model for open collaborative authoring systems based on a combination of classic integrity mechanisms from computer security and reputation systems. The model is able to accommodate a number of different integrity policies and three different policies are presented in the paper. While the model provides a reputation based assessment of the trustworthiness of the information contained in a document, the primary objective is to prevent untrustworthy authors from compromising the integrity of the document. In order to determine the effectiveness of the proposed integrity model, we present an attacker model for open collaborative authoring systems, which allows us to calculate the vulnerability of a given document based on the fraction of malicious authors in the system.

1 Introduction

Collaborative authoring systems which support an open and dynamic population of authors, such as the Wiki [1], have become increasingly popular over the past couple of years. Large pieces of documentation, such as the Wikipedia [2], have been compiled using this type of technology and the Wiki technology has become an indispensable part of many computer supported collaborative work (CSCW) tools that support a distributed user base. While it may be argued that collaboratively authored documents will never have the same authority as a traditionally edited work [3, 4], the Wikipedia project has demonstrated the benefits of this approach by compiling a comprehensive and largely accurate encyclopedia from the contributions of individual people located around the world. However, the Wikipedia has also exposed one of the weaknesses of collaborative authoring, which is that malicious or incompetent users may compromise the integrity of the document by introducing erroneous entries or corrupting existing entries, e.g., Jimmy Wales, the co-founder of the Wikipedia, claims to receive 10 emails every day from students who failed their courses because the information cited from the Wikipedia turned out to be wrong [5] and public figures sometimes find that the entry describing them in the Wikipedia has been modified to defame them [6, 7]. Despite the contested findings in Nature [8], which found that the quality of information in the Wikipedia was almost as high as the Encyclopdia Britannica,[1] it appears obvious that a general mechanism to improve the quality of documents produced in open collaborative authoring systems is needed. The quality of a collaboratively authored document is determined by a few simple properties, i.e., is the document complete, correct and unbiased. We have previously argued that these properties correspond to the properties ensured by existing integrity mechanisms in computer security [10], so we intend to leverage this work by designing an integrity mechanism for open collaborative authoring systems. Most data protected by an integrity mechanism, however, have well defined syntax and semantics, whereas the syntax and semantics of collaboratively authored documents are difficult to define. This means that existing integrity mechanisms cannot be used directly. The obvious answer to this problem is to rely on feedback from the users, i.e., some reputation system similar to the ones used by Amazon [11], eBay [12] or the "WOT" plugin for Firefox [13]. Relying on external feedback corresponds to the approach, which is already used in a wiki (cf. Section 2.2), where other authors may revert a document to an earlier version. Reputation systems[2] have previously been proposed as an effective means to assess the quality of information from uncertain sources [15, 16, 17, 18, 19], but they only help automate detection of undesirable content and are generally unable to prevent undesirable content from being introduced into the document. We therefore propose a combination of reputation systems to assess the quality of collaboratively authored documents and traditional integrity mechanisms to prevent unknown or untrusted users from modifying the documents in the collaborative authoring system. The mechanism automatically assigns a "quality rating" to every document in the system, based on the reputation of the last user who updated the document. In order to enforce integrity, we want that only users with a similar or higher reputation than the past user will be able to modify the entry. This means that users with a poor reputation will be unable to update most of the documents in the systems, but more importantly that documents that have a high quality rating may only be modified by the few users who have an equally high reputation.

[1] The Encyclopdia Britannica issued a 20 page response [9] to the article in Nature, which questions the methods used in the study published in Nature.

[2] A good survey of reputation systems was published by Jøsang, Ismail and Boyd [14].

The structure of the rest of the paper is as follows. We start, in Section 2, with a definition of our model of an open collaborative authoring system and identify important properties of such systems. Section 3 presents a short overview of the most important integrity mechanisms developed in the context of computer security. We propose our integrity model for open collaborative authoring systems in Section 4. This model is based on a document review process, which is described in Section 5 and an evaluation of the proposed model is presented in Section 6. Directions for our future work are outlined in Section 7 and our conclusions are presented in Section 8.

2 Open Collaborative Authoring Systems

Systems that allow online communities to author and publish collaborative work can be organised in different ways. We present an outline of a generic architecture in the following, which defines our terminology and allows us to identify different properties of such systems.

2.1 System Model

An open collaborative authoring system (OCAS) is defined by a backbone server network and a set of client machines that are used by the authors to access the documents. This architecture is illustrated on Figure 1.

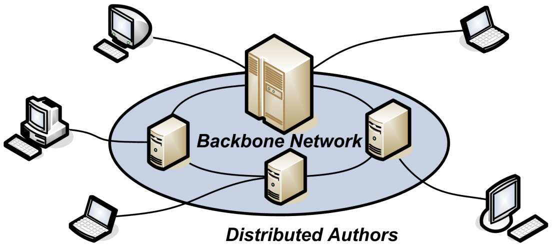

Fig. 1. System Model

The backbone network, which may consist of a single server, stores all data and mediates users' access to documents including enforcement of access control policies and synchronisation when multiple authors simultaneously wish to update the same document. The backbone network infrastructure may either be universally accessible or it may belong to a closed community, such as a company's internal intranet, which limits the set of clients that are able to access the infrastructure.

The client machine runs the necessary software to allow a distributed set of authors to read and write documents stored in the backbone network. In many cases, this software is simply a web-browser, which allows the client to interact with the system that runs on a web-server in the backbone network. Authors may be required to register an identifier that is used to identify the contributions of the different authors in the system.

2.2 System Security Model

Open collaborative authoring systems often allow any user who can access the system to create an account (possibly anonymously) and start creating new or editing existing documents. The most popular open authoring technology is probably based on Wiki technology, so we focus our analysis on policies and mechanisms developed for the wiki.[3] The basic philosophy behind a wiki is that everyone should be allowed to edit everything, but that it should be easy to restore the document to its prior state if the modifications are considered undesirable. The traditional security process is based on *prevention, detection and response*, where security mechanisms are introduced to prevent unauthorised access to protected resources, auditing procedures and intrusion detection systems are introduced to detect unauthorised use of the system and a combination of automatic and manual procedures are used to stop unauthorised access and return the system to a consistent state. Applying this process to the wiki philosophy, we see that there are few mechanisms to prevent malicious or accidental modification of a wiki page; detection is left to the users and the only means of response is to restore the previous version of the page.

Authors in wiki-style systems are often required to create an account before they can edit pages, but this is not an essential requirement. Moreover, authors may create multiple accounts and it is generally not possible to identify the person who registered a particular account. This means that the primary means of authentication is the password that the author provided when he created his account, i.e., all information needed to create an account may have been provided by an otherwise anonymous user. This results in a lack of accountability, which was exploited by unscrupulous users in the examples listed in the Introduction. The only effective means of recognising users in most wikis is therefore the IP-address of the client machine, which allows malicious users operating from a fixed Internet address to be identified and blocked, but users operating from a machine with a dynamically allocated IP-address and those who operate from public access terminals, e.g., public libraries or Internet cafés, cannot be blocked.

The threshold to enter a wiki is often relatively low, e.g., all users are granted the same access privileges, so the cost of discarding a compromised account identifier and create a new account is equally low. This means that wiki-style systems are extremely vulnerable to the Sybil attack [20], where an attacker may create a new identifier for every attack. It is difficult to completely avoid the Sybil attack without logically centralised identity management, e.g., a public key infrastructure (PKI), but it is possible to reduce the problem by increasing the cost of generating new identities. This approach would be similar to HashCash [21] and “proof-of-work” systems [22] used to fight Spam email.

[3] Although our definition of an open collaborative authoring system is not limited to existing wiki-style systems, we sometimes use the terms wiki and wiki-style systems in place of the much longer open collaborative authoring systems.

3 Integrity Mechanisms in Computer Security

Integrity mechanisms are designed to prevent corruption or destruction of data managed by the system. This means that mechanisms should be in place to prevent malicious users from tampering with data directly and to ensure that data is kept coherent, e.g., all modifications of data should be through well-formed transactions [23].

One of the first models to address the problem of integrity was proposed by Biba, who defined an integrity model [24] analogous to the well known mandatory access control model defined by Bell & LaPadula [25]. The model divides subjects (processes running on the behalf of named users) and objects (stored data manipulated by the processes) into different integrity classes and defines the following two security properties:[4]

1. Simple security property, which states that a subject at a given level of integrity may not read an object at a lower integrity level (**no read down**).
2. $*$ (star) security property, which states that a subject at a given level of integrity must not write to any object at a higher level of integrity (**no write up**).

The model prevents flow of information from low integrity documents into high integrity documents. The subjects defined by the Biba model correspond to the browsers used to update the wiki and the objects of the Biba model correspond to the documents in the wiki. Most operating systems, in common use on the Internet, do not distinguish between the access rights of different processes started by the same user, e.g., a user may freely *cut-and-paste* between all the GUI-windows on the monitor, so there is no reason to distinguish between a process and the user who launched it, i.e., a subject in the Biba model corresponds to an author in the OCAS.

The Biba model works well when the security mechanism has complete mediation, i.e., every access to every object must be checked for authority. In wiki-style systems, however, authors may have several programs open at the same time, so it will be practically impossible to enforce the simple security property, but we believe that the $*$-security property may prove useful.

Another aspect of the Biba integrity model which we believe will prove useful is the concept of a “low-watermark,” which changes the integrity class of the subject to the integrity class of the lowest object accessed by the subject. As we mentioned above, we do not expect complete mediation, so we will not be able to correctly update the integrity level of subjects, instead we propose to interpret the low-watermark policy with respect to the object, which means that we change the integrity class of the document to the integrity class of the subject with the highest integrity class who accessed the object.

Another influential integrity model was defined by Clark and Wilson who realised that integrity is more important than confidentiality in many commercial and government applications. The Clark-Wilson integrity model [23] is not explicitly

[4] These security properties are the inverse of the properties defined in the Bell & LaPadula model, which focus on confidentiality instead of integrity.

based on integrity levels, but is instead based on the notion of well-formed transactions that transform a system from one consistent state to another. This means that integrity is not considered an inherent property that can be used to label subjects and objects. Instead, integrity is defined by the relationship between certain object and it can be enforced by imposing constraints on the operations that subjects can perform on objects. This model is not directly applicable in collaborative authoring systems, because it appears difficult to define criteria for consistency of text documents and few general text editing methods can be considered well-formed transactions.

The low-watermark integrity policy has recently been explored in the context of commercial off-the shelf software [26]. The LOMAC integrity mechanism is implemented as a Linux kernel extension, which supports low water-mark integrity policies in a standard Unix system. The default policy only defines two integrity levels for data: high for system files and devices, except the network interface card, and low for user data. Integrity policies with more integrity levels can be defined when the system is installed. Processes inherit the integrity level of the user who started them, but if they access data with a lower integrity level, their own integrity level is demoted to that of the data.

4 security in Open Collaborative Authoring Systems

In order to improve the quality of documents in an open collaborative authoring system, we define an integrity mechanism that limits the set of documents that an author can update. The integrity mechanism is based on two basic integrity models: the *static integrity model* and the *dynamic integrity model*, which capture respectively the static and dynamic properties of integrity control.

4.1 Static Integrity Model

The static integrity model defines whether a registered author should be allowed to edit a particular document based on the quality of the author's previous contributions and the estimated quality of the document. As all users should be allowed to read all information in an open collaborative authoring system, the static integrity model defines the actions of the *reference monitor*, i.e., it defines the access control model.

All authors must have an identifier[5] (possibly a pseudonym) which will allow the system to recognise authors and attribute them with a quality confidence value (QCV), which indicates the normal level of correctness, completeness and lack of bias in documents by that author, i.e., it encodes the reputation of that author. Similarly, each section of the document must have an integrity level (IL) associated which it, which indicates the level of correctness, completeness and lack of bias in that particular document. There is an obvious relationship between the QCV of an author

[5] We do not require that authors have unique identifiers, so an author may have multiple identifiers but we do expect that authors do not share identifiers.

and the IL of the (sections of) documents that she has authored,[6] because authors of documents with a high IL must have an equally high QCV to ensure that the quality of documents is preserved and documents edited by authors with a high QCV are likely to improve in quality, so the IL of the edited document should be equally high. The IL of a document is determined partly by the QCV of the authors who have contributed to the document and partly by feedback from registered authors in the system. This means that the integrity level (QCV) of the last author to edit a document determines the current integrity level (IL) of that document. The integrity label of the document is modified to reflect the integrity label of the author, which is the opposite of the low water-mark policies described in Section 3. However, we believe that it is reasonable to assume that authors who have a history of writing complete, correct and unbiased documents are likely to continue in that style, so new documents edited by such authors will benefit from their involvement, i.e., the document will be raised to the high level of the author.[7] The user feedback mechanism allows documents to be promoted beyond the level of the authors who have contributed to the document as defined by the dynamic integrity model described below.[8]

Formally, the static integrity model defines the concepts of *authors*, *documents*, *quality confidence values* and *integrity levels* as:

A is the set of identifiers of authors who have registered to use the system.

D is the set of documents that are managed by the system.

I is a totally ordered set of integrity levels, such as $\{\text{low}, \text{medium}, \text{high}\}$ or $[0, 4]$, with the ordering relation "$\leq$" defined in the usual way, e.g., "low $\leq$ medium $\leq$ high" and "$0 \leq 1 \leq 2 \leq 3 \leq 4$".[9]

We also define two functions $qcv(a : A) : A \rightarrow I$ and $il(d : D) : D \rightarrow I$ which allow us to compare the QCV of an author directly with the integrity level of a document using the total order on I:

$$qcv(a : A) = \{\text{quality confidence value of } a\},$$
$$il(d : D) = \{\text{integrity level of } d\}.$$

Finally, we define the predicate:

$$can\_edit(a : A, d : D) = \{\text{'1' iff } il(d) \leq qcv(a)\},$$

[6] Without loss of generality we consider documents with a single author in the following, but the integrity model is trivially extended to smaller textual units, such as sections or paragraphs, so that documents with multiple authors may be encompassed by the model.

[7] In order to simplify our presentation, we assume that all documents in the wiki relate to the same subject area, so an author is equally competent to edit all documets. The model is easily extended to encompas documents with different subjects by introducing a classification of documents according to subject and maintaining separate QCV for each subject that an author contributes to.

[8] The user feedback mechanism should also allow the IL of a document to be lowered, if it turns out that the trust in the author was misplaced.

[9] The ordering relation "$\leq$" is similar to the *dominance relations* defined by the Bell & LaPadula and Biba models.

which returns '1'[10] if the author is allowed to edit the document.

Intuitively, an author should only be allowed to edit a document if her reputation is higher than the current quality of the document, so authors should only be allowed to modify documents when the can_edit predicate is '1'. This prevents information generated by "low integrity" authors from entering "high integrity" documents, which corresponds to the $*$-property in the Biba model. Together with a low watermark policy defined by the dynamic integrity model below, the $*$-property ensures that the integrity label of a document can only increase.

Finally, we do not associate a particular semantics with the different integrity labels and we do not prescribe a pre-defined number of labels in the model.

4.2 Dynamic Integrity Model

The dynamic integrity model controls the modification of quality confidence values and integrity levels based on the *integrity watermark model* outlined above and a *document review model*, which controls the explicit promotion and demotion of documents based on feedback from the other authors in the system. These models are described in greater details in the following.

Integrity Watermark Model

As mentioned above, authors with a high QCV are likely to improve the quality of documents with a lower IL, so we define a model, based on the classic watermark model, to increase the IL of documents that have been modified by good authors and increase or decrease the QCV of authors who have contributed to documents that change their IL as part of a document review (cf. Section 5.)

We define the function $\mathcal{E}(a : A, d : D) : A \times D \rightarrow D$, which is invoked when author a wish to edit document d. The changes to the IL of d are not represented in the signature for $\mathcal{E}$, because they are really side effects of editing the document. However, since the focus of this paper is on security, we only considered these aspects in our definition of $\mathcal{E}$.

$$\mathcal{E}(a : A, d : D) = \begin{cases} il(d) := qcv(a), \text{if } can\_edit(a, d), \\ \textit{security violation}, \text{ otherwise.} \end{cases}$$

The integrity watermark model allows good authors to raise documents to their own integrity level, but it has no effect on the QCV of the other authors who contributed to the document. We therefore introduce a mechanism that allows an author to improve her QCV by submitting a number of documents, which she has contributed to, for a *document review* by other authors (some of these authors *must* have a higher QCV in order to protect against Sybil attacks.) This review mechanism implements the document review model defined below.

[10] We generally use the normal programmer's convention of representing the boolean values 'true' as '1' and 'false' as '0'.

Document Review Model

Any author who has contributed to a document can request a document review, which will determine whether the IL of the document should be increased as a result of the modifications that have been made since the last document review; this is known as a *promotion* of that document. If the documents are promoted, the principal author of the documents will also be promoted. The *principal author* may be defined in different ways, e.g., it may be the author who contributed most modifications since the last promotion, the author who contributed the latest modifications, the contributing author who proposed the promotion of the document or any combinations of the above. It is also possible that modifications reduce the quality of a document, so it must be possible to decrease the IL of the document; this is known as a *demotion* of that document. All users who can edit a document, can also request a document review, which will determine whether the document should be demoted. This means that all authors who can_edit a document may request a demotion review, while only authors who have actually contributed to a document may request its promotion. We believe that this strikes a reasonable balance between integrity of the document and the threat of denial of service through spurious document review requests. If a document is demoted, the QCV of the principal author will be reduced accordingly. In this paper, we propose to promote/demote the author who contributed most to the document, but this may not be appropriate for all collaboratively authored documents, so we aim to study the effects of different promotion-/demotion strategies on the dynamics of collaboration in future work.

As already mentioned, wiki-style systems are vulnerable to Sybil attacks, where an attacker registers an author identifier, which is then used to corrupt data in the system. When the culprit is found and the identifier has been banned from the system, the attacker registers a new identifier and continues to corrupt the system. Moreover, the attacker may simultaneously register multiple identifiers and use this to orchestrate collusion among seemingly unrelated authors. It is therefore important that new users are introduced at the lowest integrity level, so that they are unable to corrupt documents that have already been reviewed or improved by good authors. It is equally important to ensure that an attacker who registers many new author identifiers is unable to improve the QCV of any of these identifiers through the document review mechanism, so we require that authors with a higher QCV is involved in the review of each document. Moreover, the work required to raise a sufficient number of author identifiers to a QCV level that would allow an attacker to control promotions from lower levels must therefore be high (cf. Section 6.2).

The set of integrity levels (I) with its total ordering, defines a natural hierarchy among authors and documents. As mentioned above, new authors must be introduced at the lowest integrity level and are initially only able to create and contribute to documents as this level. In order to be promoted, they have to submit high quality documents for a document review. By requiring that authors with a higher QCV participate in the review, we help protect against the Sybil attack described above and each level in the hierarchy that is involved in the review increases the robustness against this form of attack. The number of levels in the hierarchy depend on the number of elements in I and the number of levels that must be involved depend on the desired balance between document integrity and the workload needed to review documents. As there is no protection against malicious authors at the initial level, we believe that the number of levels involved should be at least 3. Moreover, we observe that the top levels in the hierarchy cannot be protected by the levels above, so the distance from top to bottom must be sufficiently large to increase the difficulty for an attacker to control the highest levels by getting a sufficient number of malicious authors promoted. This is the reason that we propose at least 5 levels of authors, so that there must be at least 5 different integrity levels in I. In the following we describe a document review system with exactly 5 levels, numbered from '0' to '4'.

5 Document Review

The following document review process is defined to protect the integrity of documents in an OCAS. The documents are divided into several integrity classes according to the integrity models defined above and each level L_i of the system is characterised by an index i. We assume that each level in the hierarchy contains Λ_i registered authors and we assume that z_i of these authors are malicious and in collusion with each other. For the purpose of our discussion, we assume that only malicious authors will act improperly during the reviewing session of a document. During a document review, r_i of the Λ_i authors are randomly selected (without any bias) in order to review the document submitted by a user of the same or a lower level. Each of these reviewers determine whether they believe that the document should be promoted/demoted and return their verdict to the document review system. A certain majority of reviewers at a given level L_i must approve the quality of the submitted document, so that it will be accepted by all level Λ_i users. We can call this majority τ_i. Here, we can distinguish between a simple system, and a weighted one. In the weighted system, the vote of an author at level L_i may have a positive weight of w_i depending on her past performance. In the following, we limit our analysis to the simple review system, where all authors votes are weighted equally.

We define a set of numerical integrity levels where the lowest level is 0 (L_0) and that the highest level is 4 (L_4) with the total order defined by '$\leq$'.[11]

We consider $\partial_j(d)$ to be the judgement of reviewer j for the document d. Judgements can take two values: rejection or approval, '0' or '1'. In order to reason about the quality of judgements, we assume that the quality of a document can either be poor (in this case the article should be rejected and we write "$d = 0$"), or high (in this case the article should be accepted and we write "$d = 1$"). There are thus two different types of mistakes that a reviewer can make: A reviewer can either approve an article with low quality, or reject an article with high quality. These two errors can be expressed as $\partial_j(d = 0) = 1$ and $\partial_j(d = 1) = 0$, respectively.

We continue our study by defining the overall decision of the document review system upon a reviewed document. When a document at one of the three lower levels

[11] There is a simple mapping from any totally ordered set of 5 integrity levels to the set defined above.

(0, 1, 2) is reviewed, the reviewers are selected from the same level as the document and the two levels above. When an article of level 3 is reviewed, reviewers are selected from the same and the higher level (level 3 and level 4). When an article of level 4 is reviewed, then all reviewers are selected from the same level.

Let $\mathcal{D}_i(d)$ be the combined judgement of the reviewers at level L_i about a document d. Then $\mathcal{D}_i(d)$ can take two values: rejection or approval, which we write '0' or '1' respectively. The judgement at a specific level depends on the judgement of the subset $\Lambda_{R_i} \subseteq \Lambda_i$, that includes r_i randomly selected reviewers from Λ_i, as:

$$\mathcal{D}_i(d) = \begin{cases} 1, \text{if } \sum_{j \in \Lambda_{R_i}} \partial_j(d) \geq \tau_i \\ 0, \text{otherwise} \end{cases} \tag{1}$$

We now define $D(d)$ as the final decision (judgement) of the reviewing session of the system about a document d; $D(d)$ takes two values: rejection or approval, i.e., '0' or '1'. We are particularly interested in documents submitted by users of the three lowest levels (i.e., L_0, L_1, and L_2,) because we are able to define many different policies governing the final decision $D(d)$ depending on the outcome of the judgement of the reviewers at each level $L_i : D_i(d)$. The definition and analysis of these policies are, however, beyond the scope of this paper, so we limit our analysis to a single simple policy, which we will call Π_1. This policy requires the acceptance of the document by a simple majority of the levels involved in the review, which means approval by any two of the three sets of reviewers from levels L_i, L_{i+1}, L_{i+2}, i.e., we require that at least two of the $D_i(d), D_{i+1}(d), D_{i+2}(d)$ have the value '1'. We express this policy in Equation 2.

$$D(d) = (D_{i+2}(d) \wedge D_{i+1}(d)) \vee (D_{i+1}(d) \wedge D_i(d)) \vee (D_i(d) \wedge D_{i+2}(d)) \tag{2}$$

When it comes to documents submitted by authors at the two highest levels (that is L_3 and L_4,) we have to follow a different approach for two reason. First of all, there are not "many" levels that are higher, as mentioned above and secondly, these levels *must* be guaranteed to contain high quality documents and have honest reviewers, because they guarantee all the lower levels in the system.

Continuing our analysis, when a document at L_3 is reviewed, we define a simple policy for the outcome of the review process ($D(d)$,) which requires the acceptance of the document by both sets of reviewers from levels L_3 and L_4. Thus, we require that $D_3(d)$ and $D_4(d)$ have the value '1'. We express this policy as: $D(d) = D_3(d) \wedge D_4(d)$. Finally, when a document at L_4 is reviewed, then we require its the acceptance by reviewers from L_4 only, so we have: $D(d) = D_4(d)$.

5.1 Operation Considerations

The document review system relies on voting, which means that the system must be populated in order to work. In the following, we examine how a system working according to our model may be bootstrapped and how more authors may be added to the system.

All wiki-style systems have an initial author, or an initial group of authors, who decide to decide to start and run the system. Part of the system configuration consists of creating an author identifier for the initial user at each level from L_0 to L_4 – if a group of authors decide to start the wiki, they may distribute these author identifiers among themselves. Having at least one author defined at each level ensures that it is possible to establish the majority of authors at each level that is required by the document review process. It is important that this initial set of authors are never promoted, so we avoid the risk of empty levels in the hierarchy.

When a new author registers in the system, then he is a level L_0 user. After significant contributions, the authors should be promoted to the next level, e.g., from L_0 to L_1. The required contribution is a number of constructive articles. The term constructive articles refers to high quality articles that are accepted in the system after a document review.

Let α_i denote the effort (in terms of successfully promoted documents) needed for a single author to be promoted from level L_{i-1} to level L_i, where level L_0 represents new authors who have not yet contributed to the system. It must be easy for new authors to start contributing to the system, so we simply require them to register, i.e., $\alpha_0 = 0$. The definition of the effort required for the other values of $\alpha_i (0 < i \leq 4)$ is an important parameter in the security of the system, which we will examine shortly in the evaluation. Furthermore, an author would be demoted after a number of destructive entries (β_i). The term destructive articles refers to low quality articles that are demoted in the system after a reviewing session. The index i in β_i means that an author is demoted from level L_i to level L_{i-1}. The definition of the disruption required to demote an author (β_i) is another important security parameter. We believe that the definitions of both α_i and β_i are likely to depend on the OCAS, both the integrity requirements for the documents and the size of the system. Moreover, it is possible that these should be dynamic parameters, so that it becomes more difficult to get a document promoted (α_i increases) if there have been relatively many unsuccessful promotion attempts within a short period of time (this may be an indication that someone is attacking the system). A full analysis of α_i and β_i is, however, beyond the scope of this paper and we leave this for future work.

6 Evaluation

In the following we examine the system's ability to resist attacks from malicious users. We assume that the necessary security mechanisms, such as authentication of registered users and the access control mechanism that implements the static integrity model have been correctly implemented, so we focus on the possible damages caused by registered, authenticated and authorised users.

6.1 Attack Model

The reviewing procedure for an article is the following: Every time an article d is to be reviewed, then r_i members of a level L_i are selected as reviewers, in a random manner. A level has Λ_i users, and z_i of them are malicious. Eventually, the probability that the reviewing process will accept a poor quality article is equivalent to the probability that there are enough malicious reviewers among the r_i randomly selected ones, so as to bias improperly the outcome of the reviewing process. Let this probability be p_i; we can estimate it using Equation 1 and the hypergeometric discrete probability distribution. According to the policy Π_1 defined in Equation 2, the simple majority of reviewers at each level who must approve the submitted document (τ_i) is actually a "threshold" for the malicious reviewers in the system. In our study, we are interested in scenarios with at least τ_i selected malicious users, because that is the only way colluding malicious authors may control the document review. The number of malicious authors who may participate in a document review is only limited by the number of selected users r_i and by the total number of malicious users z_i. We are therefore able to estimate the probability (p_i) that a set of z_i malicious authors may influence the decision of the document review process as:

$$\begin{aligned} p_i &= \text{Prob}\{D_i(d) = 1 | d = 0\} \\ &= \text{Prob}\{\sum_{j \in \Lambda_{R_i}} \partial_j(d) \geq \tau_i | d = 0\} \\ &= \text{Prob}\{\tau_i \text{ or more of the } r_i \text{ randomly selected reviewers are malicious } | d = 0\} \\ &= \sum_{k=\tau_i}^{min(r_i,z_i)} \frac{\binom{z_i}{k} \cdot \binom{\Lambda_i - z_i}{r_i - k}}{\binom{\Lambda_i}{r_i}} \end{aligned}$$

The hypergeometric probability distribution is useful when we have to estimate the probability of having exactly k malicious users among the r_i that are randomly selected, provided that level L_i has z_i malicious users.

In order to illustrate the effect of p_i in a real system, we consider a small wiki with 32 registered authors at each of the levels. In order for an attacker to influence the decision of the review process, he must control a certain number of authors. For the sake of this experiment we consider the total number of registered authors to be fixed; this is not unrealistic because we could implement a collusion detection system that raises an alarm if many new users are registered within a short period of time.We have analysed three different scenarios, which allows us to examine our policy Π_1 for three different sets of parameters. In the first scenario (Scenario 1), Π_1 requires that half of the authors at a given level participate in the review (i.e., $r_i = 16$) and we require a simple majority among the participating reviewers (i.e., $\tau_i = 8$). In the second scenario, we only require a quarter of the registered authors at a given level to participate and we still require a simple majority among the participating voters (i.e., $r_i = 8$ and $\tau_i = 4$). In the third scenario, we again require a quarter of the authors at a given level to participate, but we now require a three quarter majority among the participating voters to approve the document (i.e., $r_i = 8$ and $\tau_i = 6$). We have calculated the number of malicious users z_i that are required at a single level in order to have a certain probability of influencing the outcome of the document review. The result of these calculations is shown in Table 1.

	Probability of influencing decision					
	95%	90%	75%	66%	50%	33%
Scenario 1	20	19	17	16	15	14
Scenario 2	22	20	18	16	14	12
Scenario 3	28	27	25	24	22	20

Table 1. Number of malicious users that must be controlled to influence decision.

For each scenario, the table shows the minimum number of authors that must be controlled in order to have a specific probability of influencing the outcome of the document review. The table shows that an attacker must control more than two thirds of a level in order to have a higher than 95% probability of influencing the outcome of the document review *at that level.* It also shows that nearly half the authors must be controlled at a particular level in order to have a better than 50% chance of influencing that level. Finally, it shows that changing the simple majority vote to a qualified majority vote, increases the number of authors that must be controlled dramatically.

6.2 Cost of Attack

By analysing the dynamic integrity model and the document promotion scheme presented in Section 5, we can now estimate the effort required to launch an attack against the system. When, for example, an attack is launched at level L_0, and there are z_0, z_1, and z_2 malicious users in levels L_0, L_1, and L_2, respectively. If an attacker wishes to have high probability of success, he must try to have as many malicious authors as possible among the authors of the levels involved in the document review. These are the level that he wishes to attack, and quite often, levels that are higher than this. All these imply that he must "pay" a significant effort in order to promote a sufficient number of malicious authors. For the moment, we estimate the needed effort (cost) to have one malicious user in level L_i as:

$$\sum_{j=0}^{i} \alpha_i, \text{ where } \alpha_0 = 0 \tag{3}$$

Together with the success probability presented in Section 6.1, the cost of attacking a specific level allows us to calculate an estimate of the vulnerability of an OCAS according to the voting policy chosen in the document review process.

6.3 Cost of Attacking System with Policy Π_1

We are now ready to calculate the cost of an attack against a particular level of a system that implements the document review policy Pi_1.

We base our analysis on the policy specified above, which states that $D(d) = (D_{i+2}(d) \wedge D_{i+1}(d)) \vee (D_{i+1}(d) \wedge D_i(d)) \vee (D_i(d) \wedge D_{i+2}(d))$. Therefore, the cost $\mathcal{C}_0$ of an attack to Level L_0 will be estimated as:

$$\begin{aligned}\mathcal{C}_0 &= min\{z_1(\alpha_0+\alpha_1)+z_2(\alpha_0+\alpha_1+\alpha_2), z_0\alpha_0+z_1(\alpha_0+\alpha_1),\\ &\qquad z_0\alpha_0+z_2(\alpha_0+\alpha_1+\alpha_2)\}\\ &= min\{z_1\alpha_1+z_2(\alpha_1+\alpha_2), z_1\alpha_1, z_2(\alpha_1+\alpha_2)\}\\ &= min\{z_1\alpha_1, z_2\sum_{j=0}^{2}\alpha_j\}\end{aligned}$$

In a similar manner we can estimate the cost of an attack to the other 4 levels and a summary of the results is shown in Table 2.

Level	Cost of attack
L_0	$\mathcal{C}_0 = min\{z_1\alpha_1, z_2\sum_{j=0}^{2}\alpha_j\}$
L_1	$\mathcal{C}_1 = min\{z_1\alpha_1 + z_2\sum_{j=0}^{2}\alpha_j, z_2\sum_{j=0}^{2}\alpha_j + z_3\sum_{j=0}^{3}\alpha_j,$ $z_3\sum_{j=0}^{3}\alpha_j + z_4\sum_{j=0}^{4}\alpha_j\}$
L_2	$\mathcal{C}_2 = min\{z_2\sum_{j=0}^{2}\alpha_j + z_3\sum_{j=0}^{3}\alpha_j, z_3\sum_{j=0}^{3}\alpha_j + z_4\sum_{j=0}^{4}\alpha_j,$ $z_4\sum_{j=0}^{4}\alpha_j + z_2\sum_{j=0}^{2}\alpha_j\}$
L_3	$\mathcal{C}_3 = z_3\sum_{j=0}^{3}\alpha_j + z_4\sum_{j=0}^{4}\alpha_j$
L_4	$\mathcal{C}_4 = z_4\sum_{j=0}^{4}\alpha_j$

Table 2. Cost of an attack according to policy Pi_1

The table above shows us that it is important to define the cost of progressing to the next level (α_i) and the voting policies so that a high number of malicious users (z_i) is required to control a specific level (L_i.) Moreover, the document review process should be defined to ensure values of α_i and z_i that increases the cost of controlling the different levels, as seen in Table 2.

7 Future Work

The model defined and analysed in this paper makes two simplifying assumptions that would seriously impact it's use in real systems. As already mentioned, these assumptions are easily satisfied by trivial extensions to both the static and dynamic integrity models. We discuss these extensions, before we go on to suggest other directions for future work.

The first assumption is that documents are edited by a single author at the time, i.e., all modifications to a document can be attributed to a single author. This assumption is introduced to simplify the discussion of the relationship between the QCV of authors and the IL of the documents that they modify, but the model is trivially extended to cover smaller textual units, such as sections, paragraphs or even sentences. This will allow multiple authors, possibly with different QCVs, to collaborate on a single document. However, managing documents where different sections have different integrity levels raises a number of practical issues that must be addressed in a real system. First of all, we need to determine how the different integrity levels in such documents can be presented to readers in a simple and intuitive way, e.g., by using different background colours [27]. Secondly, we need to investigate the impact of different textual granularities of the integrity mechanism and determine whether there is an optimal granularity or whether the granularity should be a parameter that is configured when the wiki is installed. Finally, we need to determine the impact of a finer textual granularity on the voting protocol, e.g., will reviewers stay vigilant when they are asked to decide whether the third paragraph in a document merits promotion.

The second simplifying assumption is that all authors are assumed to only contribute to documents within their subjects of expertise. This means that we do not have to consider well intended authors who make good contributions to documents within their area of expertise, but poor contributions to documents in other areas, i.e., we only need to manage one QCV for each author. This is easily solved by incorporating a classification scheme, similar to the *Categories* found at the bottom of every article in the Wikipedia, which allows the system to identify the authors areas of expertise (those where the documents are generally promoted.)

In our presentation of the document review process, we confined our analysis to a single simple policy for combining the judgements from different levels in the hierarchy (Π_1). Other policies are easily formulated and it would be particularly interesting to explore policies that are substantially different from the simple majority rule, e.g., different (possibly dynamic) qualified majorities at the different levels in the hierarchy. The document review process considers all reviewers with the same weight, but it would be interesting to explore the effects of weighted votes where the votes of different reviewers count differently, e.g., through the use of a reputation system. This will have an impact on security, e.g., the model's ability to resist Sybil attacks, so we need to analyse the impact of introducing weighted votes on the overall security of the system. Finally, we plan to investigate the impact of run-time adaptability of the different security parameters identified in this paper.

8 Conclusions

In this paper we addressed the problem of ensuring the quality of documents in open collaborative authoring systems, such as wikis. Existing solutions primarily focus on assessing the quality of the documents themselves, either through an analysis of the documents or through an evaluation of the trustworthiness of the source of the document. While such techniques are extremely useful, they do little to prevent malicious users from corrupting the documents maintained within a community. The lack of authentication in many wiki-style systems leads to a lack of accountability, which results in lower quality of the documents. Instead of introducing a centralised authentication authority, we propose a novel integrity model where authors earn the right to modify documents by contributing to the system, which raises the cost of performing a Sybil attack.

The proposed integrity model combines existing assessment techniques with integrity control mechanisms from computer security, in order to provide quality information to the reader and prevent untrustworthy users from corrupting high quality documents. Documents are internally labelled with an integrity label, which can also be shown to the reader who will then learn something about the provenance of the document and get an idea about whether the content should be trusted. The model also also associates integrity labels with authors, which allows a system to prevent authors who have primarily authored low quality documents from modifying documents with a high quality label. The integrity mechanism is designed to ensure that the editing process does not lower the integrity of documents. The model is quite flexible and allows many different policies to be defined with respect to the set of authors selected, quorum and (qualified) majority needed to promote documents to a higher integrity label or demote documents to lower integrity labels.

We presented an analysis of the proposed model, which estimates the probability of a successful attack against a level in the defined integrity hierarchy, given a specific number of malicious authors controlled by the attacker. We also defined the concept of cost of an attack as the effort required to promote a malicious author to the desired level and showed how this may be used to estimate the overall cost of an attack on a system that implements the model. Finally, we have shown how these estimates may be used to determine some of the important security parameters identified above.

Acknowledgements

We would like to thank several people for their contributions to this paper. Our two colleagues Nicola Dragoni and Robin Sharp, who provided helpful comments and constructive criticism and in particular Petros Kaklamanis, a former student in our group, who contributed to the formalisation of many of the concepts presented in Section 5 and Section 6 of this paper.

References

1. What is Wiki. `http://www.wiki.org/wiki.cgi?WhatIsWiki`, visited 28 December 2006
2. Wikipedia, the free encyclopedia. `http://en.wikipedia.org/wiki/Wiki`, visited 28 December 2006
3. Peter Denning, Jim Horning, David Parnas and Lauren Weinstein (2005) *Wikipedia Risks*, in Inside Risks 186. Communications of the ACM, 48(12)
4. Bertrand Meyer (2006) Defense and Illustration of Wikipedia, `http://se.ethz.ch/~meyer/publications/wikipedia/wikipedia.pdf`, *Visited 15 January 2009*
5. Andrew Orlowski (2006) Avoid Wikipedia, warns Wikipedia chief, It can seriously damage your grades. In The Register, 15th June 2006
6. John Seigenthaler (2005) A false Wikipedia 'biography'. Editorial in USA TODAY, 29 November 2005

7. Maurice Chittenden (2006) Comedy of errors hits the world of Wikipedia. In The Sunday Times, 12 February 2006
8. Jim Giles (2005) Internet encyclopaedias go head to head. In Nature, December 15, 2005.
9. Encyclopdia Britannica (2006) Fatally Flawed, Refuting the recent study on encyclopedic accuracy by the journal Nature, March 2006.
10. C.D. Jensen (2007) Integrity in in Open Collaborative Authoring Systems. In Proceedings of the Joint iTrust and PST Conferences on Privacy, Trust Management and Security
11. Amazon website. `http://www.amazon.com`, *visited 28 December 2008*
12. eBay Internet auction website. `http://www.ebay.com`, *visited 12 March 2009*
13. Against Intuition, Inc.: WOT website. `http://www.mywot.com/en/wot/home`, *visited 28 December 2008*
14. Audun Jsang, Roslan Ismail, Colin Boyd (2007) A Survey of Trust and Reputation Systems for Online Service Provision. In Decision Support Systems, 43(2): 618–644
15. Ilya Zaihrayeu, Paulo Pinheiro da Silva and Deborah L. McGuinness (2005) IWTrust: Improving User Trust in Answers from the Web. In Proceedings of 3rd International Conference on Trust Management, Rocquencourt, France
16. Pierpaolo Dondio, Stephen Barrett, Stefan Weber and Jean-Marc Seigneur (2006) Extracting Trust from Domain Analysis: a Case Study on Wikipedia Project. In Proceedings of the 3rd International Conference on Autonomic and Trusted Computing, IEEE, 2006
17. B. Thomas Adler and Luca de Alfaro (2007) A Content-Driven Reputation System for the Wikipedia. In Proceedings of the 16th international conference on World Wide Web, pages 261–270
18. Mark Kramer, Andy Gregorowicz and Bala Iyer (2008) Wiki Trust Metrics based on Phrasal Analysis. In Proceedings of the 4th International Symposium on Wikis (WikiSym 2008), Porto, Portugal
19. Aniket Kittur, Bongwon Suh and Ed H. Chi (2008) Can you ever trust a wiki?: impacting perceived trustworthiness in wikipedia. In Proceedings of the ACM 2008 conference on Computer supported cooperative work, pages 477–480
20. John Douceur (2002) The Sybil Attack. In Proceedings of the PTPS02 Workshop, pages 251–260
21. Adam Back (2002) Hashcash – A Denial of Service Counter-Measure. Technical report
22. C. Dwork and M. Naor (1992) Pricing via Processing or Combating Junk Mail. In Proceedings of Twelfth Annual International Cryptology Conference, pp.139–147
23. D. Clark and D. Wilson (1987) A comparison of commercial and military security policies. In Proceedings of the IEEE Symposium on Security and Privacy, pages 184–195
24. K. J. Biba (1977) Integrity Considerations for Secure Computer Systems. Technical Report MTR-3153, The MITRE Corporation, Bedford, Massachusetts, U.S.A.
25. D. E. Bell and L. J. LaPadula (1973) Secure Computer Systems: Mathematical Foundations. Technical Report MTR-2547 (Volume I + Volume II), The MITRE Corporation
26. Timothy Fraser (2000) LOMAC: LowWater-Mark Integrity Protection for COTS Environments. In Proceedings of the IEEE Symposium on Security and Privacy, pages 230–245
27. B. Thomas Adler, Krishnendu Chatterjee, Luca de Alfaro, Marco Faella, Ian Pye, and Vishwanath Raman (2008) Assigning Trust to Wikipedia Content. In Proceedings of the 4th International Symposium on Wikis (WikiSym 2008), Porto, Portugal