

Trust Establishment in Data Sharing: An Incentive Model for Biodiversity Information Systems

Sukamol Srikwan* Markus Jakobsson† Andrew Albrecht† Mehmet Dalkilic†
Indiana University at Bloomington
Bloomington, Indiana 47408

Abstract—We describe a long-felt but largely neglected problem in conservation biology, and explain how it can be addressed using incentive mechanisms inspired by techniques in computer security and cryptography. The result is a new type of database suitable for highly distributed contributions of data, in which researchers are incentivised to submit data by the guarantees extended by a conflict resolution mechanism that allows for accurate determinations of data origination.

Keywords: biodiversity, conservation, conflict resolution, cryptographic timestamping, incentives, spam protection.

I. INTRODUCTION

Solutions to natural resource management and conservation issues are exceedingly complex. Several layers of information are involved, ranging from scientific information to political environments and human interests. Policy makers depend on scientific discovery to guide their decisions, and scientists depend on policy makers to deploy measures guided by their results. Conservation decisions, especially in developing countries, are routinely made based on an emergency risk assessment, focusing only on the issues of immediate concern. Scientists involved in such assessment efforts are limited to very tight schedules and a narrow range of conservation issues, leaving out several potential long-term effects as not assessable. If conservation biology is to mature into an effective science, the inability for scientists and policy makers to communicate needs to be overcome. Existing biological information, which is readily available in forms of scientific publications and web-based biodiversity databases, needs to be properly organized

and networked, and scientists must be incentivised to upload their results.

In many countries, there is currently a strong reluctance to share data due to fears of loss of ownership. This is particularly the case in research areas such as conservation biology, in which it is meaningful to analyze data obtained by individual researchers or research teams, and where scientific recognition is not associated with the data itself, but with the analysis. By strengthening the ownership of published data, as we propose techniques for, this may be overcome. One might argue that conservation efforts could be aided by sharing of scientific data within small and trusted groups, and without the more complex architecture we propose. However, while local conservation efforts are important and may benefit from such an approach, it would be much harder to guide cross-border efforts in this way, due to the inherently weaker trust in such settings. Global efforts, which are of great importance, could probably not rely simply tight access controls, as supported by the experience of today's systems.

The development in biodiversity information sciences has been rapid in the past five years [15], [21]. Numerous biodiversity databases and data portals have been developed and launched [2], [5], [8], [20]. The Global Biodiversity Information Facility (GBIF) [8], established in 2001, is a global biodiversity data portal that aims to promote and coordinate the compilation and standardization of the world's biodiversity data. The goal is to make it possible for global biodiversity data to be electronically accessible though collaboration between regional biodiversity units. To complement this global effort, *regional* biodiversity databases need to be established to provide the critical baseline data. Although GBIF has obtained a great amount of data (up to 1 billion records in August 2006), the data consists primarily of species-occurrence records from museum collections. Such data

* Center of Genomics and Bioinformatics. Email address sjakobss@indiana.edu.

† School of Informatics. Email addresses {markus, jaalbrech, dalkilic}@indiana.edu.

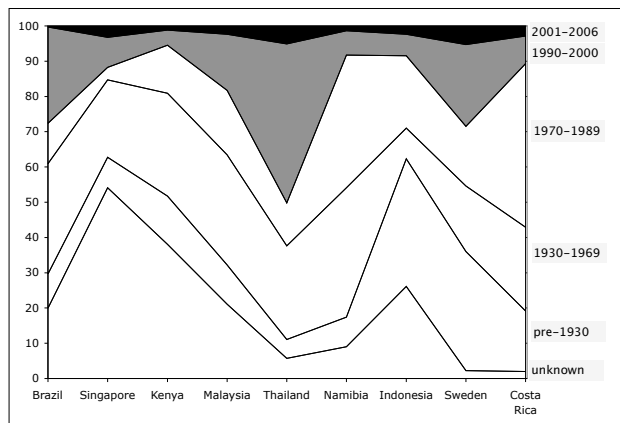


Fig. 1. The area plot shows the distribution of GBIF records from nine countries over time. The black and the gray areas represent the portion of the records that are highly relevant for conservation efforts, where the black area corresponds to the most recent data. The countries were chosen to represent (a) countries with high numbers of records in GBIF, namely Sweden (6423321 records and Costa Rica (2913126 records); (b) countries with medium number of records, namely Brazil (254524 records) and Indonesia (209825); (c) countries with low number of records, namely Thailand (74616 records), Kenya (89451 records) and Malaysia (60102 records); and (d) countries with hardly any records at all—Namibia (27827 records) and Singapore (8447 records.) Data was obtained through advanced search on the GBIF portal. Sample sizes of 116328, 41616, 42466, 38674, 15686, 37817, 25048, 5372 and 1458 were obtained for Sweden, Costa Rica, Brazil, Indonesia, Thailand, Kenya, Malaysia, Namibia, and Singapore respectively.

is typically not suitable to base conservation efforts on, given that these need to rely on recently collected data. We refer to figure 1 for a view of the severity of this aspect of the problem. Furthermore, fifty percent of the records were obtained solely from databases in the United States, the United Kingdom, and Sweden. Thus, the majority of the data can not help with conservation decisions in the countries where the efforts are most urgently needed. These are also the countries in which the reluctance to share data is the greatest, as we discuss below.

Data available through GBIF—though greatly beneficial to theoretical biodiversity research—is not useful for more applied conservation efforts. Biodiversity and conservation research efforts are still in their infancy in developing nations. Researchers in regions with rich biodiversity are still on the stage of cataloging the basic knowledge, as very little is known of many species in those regions. Informal surveys indicate that the lack of current data from developing nations is due not only to the relative technological infancy of the field in

these countries, but also due to a lack of incentives for data sharing among the researchers in question. These sentiments were echoed in several presentations at the latest GBIF symposium [14], [18], [19], but with no suggestion of how to address the problem. The incentives for data sharing in existing biodiversity information systems, such as GBIF, relate solely to the communal benefit of learning from other people’s data, and the potential of large-scale analysis. This type of systems are attractive to users who have advanced knowledge and facilities to conduct such analysis, but less so to potential users from developing countries—who often fear that researchers with better analysis opportunities may claim discoveries made from their data sets, if published.

It is clear that existing data sharing incentives do not appeal to researchers in countries most in need of the establishment of biodiversity databases. Nonetheless, it is the data from those countries, often the richest in terms of biodiversity, that are most needed to complete the global biodiversity analysis. Thus, we believe that identifying and implementing the appropriate incentives for data sharing among researchers in the developing world is crucial, however largely neglected. Computer security specialists seem not to be involved in any of the ongoing conservation efforts, in spite of the fact that computer security is a central component of establishing trust and incentives, and is also relevant in terms of privacy requirements and reputation management.

While computer security used to be a simple matter of encryption, password-based access control, and to some extent authentication, there is an increasing body of literature in which the dominating principles of computer security are used to address seemingly unrelated problems. The principles used in these efforts are associated with gaining an understanding of the *desired use* of a system, along with any potential *abuse* or *lack of use* arising from insufficient deployment motivation, and to then design the system to align incentives with the desired use. Whereas this field is still being established, it is receiving increasing attention, especially in situations where the proper incentive structure has a commercial or political impact. As a non-technical example of a commercial situation where two slightly different designs lead to vastly different results, we point to differences in initial deployment speeds (pre-1999) of cellular phones in Europe versus North America. In Europe, the monthly cost of having a cellular phone and *receiving* phone calls using this was low in comparison to the cost in North America, whereas the cost of *placing* phone calls using on a European network was relatively speaking much higher. The difference on a superficial level is merely a

matter of what services are charged for, and might seem not to be of importance to deployment rates. However, the lower perceived costs among European users resulted in an almost overnight market penetration, at which point most users started placing phone calls from their phones as well as receiving them. In our terminology, European end users were *incentivized by the design* of the charging structure to deploy the technology in question, but without any real cost to the service providers. The same general design is applicable to other situations, whether they relate to deploying technology (e.g., [9], [13]) or participating in loosely knit collaborative peer groups. The latter is a focus of this paper. Namely, using an incentive design anchored in a clear understanding of the desired—and undesired—uses of an information sharing tool for conservation biologists, we propose a system that we hope will fuel cross-disciplinary uses of collectively obtained information of conservation importance.

Before describing the actual design, it is worthwhile to dwell on the differences between database deployment for uses in molecular biology and for uses in conservation biology. While the former has seen notable proliferation, there has been very limited progress in the realm of the latter. It is important to understand the reasons for this, as they impact what approach is most suitable. First of all, it is the case that within molecular biology, large sets of data points are produced by different research institutions, but any one such set has very limited use by itself. To reach meaningful insights, much bigger quantities are needed. This creates an incentive within the society of researchers to collaborate to make progress. Moreover, the nature of the data creates another form of incentive. If we for a moment consider the individual research institution as a selfish and rational entity, then we see that there are only limited benefits in terms of competitive advantage that this entity can reap by using collectively owned data but not contributing to this communal pool. The reason for this is that if everybody else submits their data, then the difference between contributing or withholding a given set of data is fairly limited, and likely to be overshadowed by the benefits in publicity and goodwill derived from being a “good citizen”. In contrast, the typical type of data in conservation biology can be successfully¹ analyzed in relatively small quantities, such as those obtained by a single researcher or institution, while at the same time, the perceived losses of relinquishing controlling ownership may be fairly large. The reasons are, once again,

¹This is not to say that the pooling of data is not meaningful in conservation biology. Quite on the contrary: It is only when data is pooled that meaningful conservation decisions can be made.

that *somebody else* would be enabled to perform some analysis given the data in question, where this analysis would not be feasible without the data. This means that it is of importance to provide an incentive mechanism that minimizes the perceived risks of contributing data in conservation biology. In this paper, we propose a system that is mindful of this aspect of the problem.

Outline. We begin by outlining the desired technical functionality of the system (section II), followed by a description of the system architecture (section III). We then describe the detailed posting process (section IV) and the audit process (section V) used to resolve conflicts. We summarize our achievements in section VI.

II. INCENTIVE AND SECURITY MECHANISMS

There are two central assumptions that underlie all of our design. First, *a current incentive for publishing (whether in proceedings, journals, or other media) is the recognition associated with the publication. This recognition assigns an ownership of the associated insights and analyzed data.* Furthermore, *by introducing means for establishing ownership of large data sets, researchers will have an incentive to make such data sets public.* The operation of the proposed system relies on the active discouragement of a collection of unwanted behaviors, and will rest on those important assumptions. Our approach is based on the principle of time-stamping of data. As such, it provides a mechanism similar to what technical reports offer to text based material. However, a significant difference lies in the search and audit capabilities in our proposed system, which are not part of the structure for other time-stamping approaches.

In the following, we will detail the potential problems and how these are addressed by our proposed system.

a) Encouraging data sharing and resolving authorship conflicts: One of the most important roadblocks to information sharing is a common fear that other users will copy or use data without proper attribution, and that it is not going to be possible to prove such misconduct. One contribution of our proposed structure is a mechanism for cryptographic timestamping of submitted data in order to allow such conflicts to be resolved. *These cryptographic timestamps do not depend on the local system clock of a given user, cannot be forged, and can be verified by a third party.* Similar constructions are in use to audit contractual obligations, and are legally recognized as binding. While cryptographic timestamps are well understood among security practitioners, they have not been deployed in settings like the one investigated herein. Cryptographic timestamps carry the same guarantees against tampering (of time or data) as, for

example, SSL keys used within e-commerce. Anybody can verify the validity of a cryptographic timestamp. If either data, time or timestamp is tampered with, then this will become evident during the verification.

Our system will associate a cryptographic timestamp with each piece of submitted data; if any portion of the data gets modified by the authorized owner later on (e.g., for purposes of correction or augmentation), then a new cryptographic timestamp will be computed on the new data, while the old timestamp on the old data will still remain valid for the old data. We note that this will allow an audit of timestamps in situations where this is of relevance. It is important to distinguish between *authorized changes of data* (resulting in a new timestamp to be computed) and *tampering of data* (resulting in detection). Any legitimate change of data performed using the correct interface results in the former; any unauthorized manipulation of data or timestamps results in the latter.

The cryptographic timestamps will be resistant to any type of failure of machines (such as what could result from hardware damage, malware attacks, or unwanted password losses), and can be verified by anybody interested in determining when a given piece of data was generated. This is due to the fact that the verifiability of a given timestamp does not depend on the survival or availability of its issuer. In fact, using methods to link timestamped events, it is even the case that the corruption of a timestamping server does not pose a threat to already issued timestamps, as it remains impossible to backdate events or annul timestamps by an adversary possessing the secret key used to generate timestamps. This provides the individual researchers with a similar type of incentive as publication in conferences or journals: it allows for claims of when insights were gained. Cryptographic timestamps are well documented both academically (e.g., [10]) and commercially (see, e.g., [16]). Digital signatures and hash functions, which are the underlying and enabling cryptographic tools of timestamping, are also well documented, whether academically, commercially or legally.

b) Allowing fine-grained access control: It is of importance to limit both read and write access to information; each item of information² will be associated with a collection of access rights, both for read and write access. For both of these, in turn, access rights can be granted to individuals, groups (collections of groups

²An item can be made an arbitrarily small unit, but will for practical purposes be considered at the unit of information uploaded at any one time. If several items with different access rights are modified in parallel, and uploaded at the same time, then each unit will retain its initial access control level.

selected by a group leader), or the public. The access control aspects we propose are therefore very traditional, and are the core (and often the only) security component of many competing proposals. However, we augment the access control system by a reputation system (described below) that it turns into a new type of moderator mechanism to address blog spam, which is a problem of increasing magnitude. These issues are described next.

c) Discouraging undesired content: Any system that allows anybody to upload data is vulnerable to unwanted content in the form of spam, advertising, and malware (or links to any such content). Experience from public blogs suggests that once these are discovered by bad-intentioned individuals, the fraction of unwanted content quickly can surpass the fraction of desired content. This makes proactive filtering important (as opposed to removal of undesired content after complaints are filed). Moreover, the filtering decisions must be made by a trusted authority (as opposed to by common agreement among current users) in order to avoid that legitimate material is purged. This leads us to embracing the notion of a moderator role, wherein moderators can make decisions on whether submitted material is suitable or not. (This is currently practiced for many non-scientific blog applications, and in some sense also for yeast literature databases [3]. Yeast databases rely on so-called *curators* to review and categorize data; while this task is quite different from that of the moderator in our proposal, there are also structural similarities.)

We note that the moderation is not intended as a peer review, and would not be assumed to involve any decision about the merit of ideas and data; it would solely be a matter of screening whether submitted material is relevant or not. A collection of groups of moderators can be employed, where each group corresponds to one general topic of information, and each group member can make decisions of whether a given submission is to be rejected or accepted for inclusion in the database.

A novel aspect of our use of a moderator, though, is the combination of this with a reputation system to simplify a moderator's effort. Namely, in order to simplify the moderation task, each account (in turn associated with the party responsible for submitting the considered data) will be associated with a reputation; the reputation will be a function of the number of past posts, the approval rate of these, and the duration of membership. This will offer moderators guidance in their decision—the moderator interface will be designed to allow fast approval of posts from reputable users, and easy access to any submitted data. Similarly, users who have never before successfully posted may be grouped,

as many of them are likely not to have posted relevant material, but spam.

The notion of classification of submissions based on the reputation of the contributor also simplifies the task of automated screening of submitted data. This process has two benefits. As a first benefit, it can be used to identify likely spam content among data submissions made by first-time contributors (or rather: by contributors who have not previously had any data approved by a moderator.) This, in turn, simplifies the often arduous task of a moderator whose list has fallen in the hands of spammers. This is a problem of increasing magnitude given the recent development of programs whose very purpose it is to find blogs on which to post commercial (and often offensive) content. When computational cycles are available on the central repository, the same screening functions can be run on submissions with high reputation as well, to simplify detection of hi-jacked high-reputation accounts. This constitutes a second benefit of keyword-based screening. We note that account hi-jacking, related to identity theft and phishing, is a *social* as much as a *technical* problem, and cannot be prevented, but only detected. For a good overview of this general problem, we refer the reader to [12].

III. SYSTEM ARCHITECTURE

Entities. The system consists of the following types of entities: *users*, *moderators*, *clients*, *servers* and *backup storage*. *Users* and *moderators* are humans, only in different roles. They both interface with the rest of the system using machines with (at least occasional) network access—we refer to those machines as *clients*. A *server* is a computer dedicated to running the back-end of the proposed system; for simplicity we consider only one server, while practically speaking, there would—for purposes of availability and load balancing—be many. The *backup storage* is part of the back-end, and is dedicated to storing transactional data for purposes of dispute resolution. The backup storage only allows read operations and appendive write operations. Finally, the *dispute resolution engine* is a server that performs audits and dispute resolution; for simplicity, we assume it is housed with the backup storage. We refer to figure 2 for a graphical overview of entities and data flows.

Operations and data flow. Users can input data, edit or erase already approved data that they are the owners of, and submit data modification requests to update the central database (corresponding to a central repository). The add requests can be compiled at a time when the user does not have network access, and later submitted when she does, whereas edit/erase operations are (for

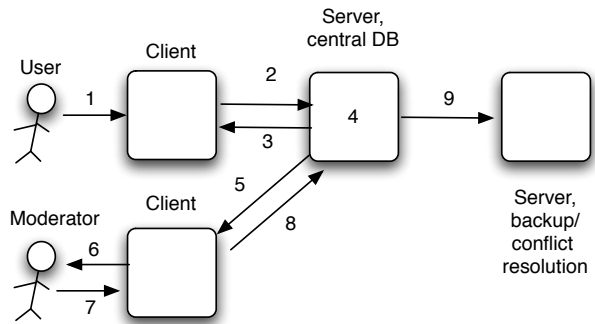


Fig. 2. General data flow: (1) user data entry; (2) submission; (3) cryptographic timestamping; (4) submission added to central database; (5) approval request; (6) data presentation; (7,8) moderator decision; (9) backup of submission. After positive moderator approval has been received by the server of the central repository, the submitted data becomes publicly viewable.

simplicity of design) initially presumed to be performed only using an online web interface. Each user needs to authenticate³ to the server when submitting requests, in order to allow ownership to be determined and to verify that the requested operation is permitted—only data owners may issue edit or erase operations. All submitted requests with correct authentication are timestamped by the server, and are immediately added to the database, but are flagged as non-viewable. As soon as the timestamps are generated, they are sent to the user who submitted the corresponding request, and are added to the database of the server. The most recent timestamp is kept on the server; all timestamps are sent to the backup storage. The submission of a request also incurs the selection of a moderator, who would receive notification of a new submission, along with a brief report on its likely trustworthiness—this does not relate to scientific quality, but merely to the likely absence of spam and other unwanted material. Once a request is approved by a moderator, the owner would be notified, and the corresponding data in the central repository is flagged as viewable. All users are given read access to viewable portions of database for searches and knowledge discovery. One or more users may request a dispute resolution to be initiated; after this has been carried out, the users involved are notified of the outcome, and provided with evidence supporting the same.

³This requires a previous setup matching the type of authentication; if password authentication is employed, then the setup would merely be to record the user name and her password. An additional step involving having to solve a CAPTCHA [1] during the authentication step may be used to reduce the number of unwanted posts by spam robots.

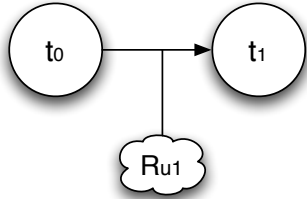


Fig. 3. Simplified view of the relation between identifiers and events; the initial timestamp identifier, t_0 , and the first event submission, R_{u1} , are used to compute the identifier t_1 , using the function $hash$. Future events are chained in with newly computed identifiers in the same manner.

IV. DETAILED POSTING PROCESS

Denotation. The requests that the user u submits consist of a sequence of SQL commands to add, edit, and delete entries in the central database. We let $R_{u\omega}$ be a submission by user u . Here, ω is a global counter maintained by the server of the central repository, and used to count both user submissions and global events. (As for any cryptographic timestamping, the global events are entered to relate the ordered submissions to real-world events, which in turn ties order to time.) Associated with each event ω , there is an identifier t_ω , which is a 160-bit pseudo-random element computed as $hash(aux, t_{\omega-1})$, where $hash$ is a cryptographic hash function, like SHA-256 [6], and aux is auxiliary data to be detailed below. The server maintains a database that is updated with all valid requests $R_{u\omega}$, and whose state after the application of $R_{u\omega}$ is referred to by \mathcal{D}_ω . This is shown in a simplified manner in figure 3.

Verifying the Validity of a Submission. To verify the validity of submission, the user authentication⁴ is verified. Submissions $R_{u\omega}$ associated with a correct user authentication are verified to be valid, i.e., not to contain any edit or delete operations associated with data that user u does not have write access to. Invalid submissions are refused, and valid submissions are timestamped and sent to the moderator.

Computing the timestamp. To compute the timestamp on $R_{u\omega}$, the server first computes the identifier $t_\omega = hash(aux, t_{\omega-1})$, where $aux = hash(R_{u\omega})$. We note that the double hash application is unusual but carries benefits as it allows the timestamps of a sequence of events to be verified without having to transmit the full event description, but only a digest thereof – where the digest corresponds to the value aux . The identifier t_ω is used both to compute the next timestamp to be

⁴The choice of user authentication approach is an orthogonal issue to the aspects treated herein, and will not be detailed due to the page limit.

issued, and to compute a digital signature s_ω on t_ω , and therefore indirectly on $R_{u\omega}$ as well. Here, any form of digital signature, such as RSA [17], DSA [7], or similar, may be employed; the choice of building block is a matter of efficiency and security trade-offs, and is beyond the scope of this paper. The value s_ω is sent to the user u .

Moderation and Update. After a moderator has approved a submission $R_{u\omega}$, it is applied to the central database. We note that submissions $R_{u\omega}$ will often be approved by the moderator out of order, given that the moderation process introduces different delays for different submissions. It will often be the case that it is possible to apply requests in their entirety out of order; when this is not the case, such requests, or parts thereof, will be kept pending. For denotational simplicity, we do not reflect this in the way we refer to the database, which we refer to by \mathcal{D}_ω after it has been updated with $R_{u\omega}$.

Storage. The server of the central repository stores the database \mathcal{D}_ω and the most recent identifier t_ω , along a set of links $L_{\omega'}$ to the backup storage, each one of which links an entry of \mathcal{D}_ω to the submission request $R_{u\omega'}$ causing the entry to be created. Edit and erase commands do not affect these links. In addition, the server of the central repository stores the secret keys needed to generate digital signatures, and data needed to verify user authentications. The backup database stores⁵ the entire sequence of tuples $(t_\omega, R_{u\omega}, s_\omega)$, along with a description of any global event used to anchor the timestamp sequence in real events.

V. DETAILED AUDIT PROCESS

An audit is a process that is initiated to settle a dispute about the order of given events relating to posting material to the database, or otherwise making related information public. It allows the generation of cryptographic evidence that has the potential of holding up in court⁶ and which therefore should also inspire the confidence to determine who is at right in communities not relying of legal arbitration. We begin by describing how to order two or more conflicting database add requests made by disputing researchers, and later describe how an order between database items and “real world” publications can be achieved.

⁵Note that it is not necessary to store a submission $R_{u\omega}$ that was turned down by a moderator; however, it is necessary to store the hash image $hash(R_{u\omega})$ of such a submission, as subsequent timestamps are functions of this quantity. Storing the hash image constitutes a slight efficiency improvement, given that these values are only 128 or 160 bits long, depending on choice of function.

⁶Digital signatures are not considered evidence in all jurisdictions, but are increasingly accepted as such.

- 1) **User-driven identification of conflicting material.** As a first step, the parties of the dispute need to identify entries in the database for which they would like to establish an order of first appearance; an arbitrary number of entries may be selected herein.
- 2) **Server-side ordering.** For each item input by the users, the conflict resolution engine queries the server of the central repository to obtain the link $L_{u\omega}$ associated with this entry. The associated list of submissions $R_{u\omega'}$ is generated and chronologically ordered. We denote ω_1 the first such item or event, and ω_n the last such event. The identifier t_{ω_1-1} is output, along with the hash of each submission in the database that lies in the interval between ω_1 and ω_n . This list also includes identifiers not associated with any of the links involved in the dispute. Along with each such item, the corresponding hashed request $hash(R_{u\omega})$ is output. In addition, all the submission requests $R_{u\omega}$ associated with disputed elements are output, along with the signatures s_ω that accompany them.
- 3) **Client-side verification.** Upon receipt of these values, the client machines of the users in the dispute compute and verify the entire chain of identifiers t_ω in the interval $\omega_1 - 1 < \omega \leq \omega_n$. This is done by re-computing these from their associated input information, e.g., the previous identifier and a hash of the submission associated with the identifier. Next, all the submissions $R_{u\omega}$ that were output are hashed, and the result compared to the hash-values already verified to be associated with the chain of identifiers. This certifies the claimed chronological ordering of the events.
- 4) **User interpretation.** Given this chronological ordering, the users involved in the dispute can analyse the series of ordered submissions to determine who made a given assertion first. This is the only part of the process that has any subjective nature at all; whereas this step may potentially be automated onwards, the best approach to do this is an open research problem, although with some interesting recent progress [4] for text-based data.

To relate database entries to real-world events, the user identifies the date of the real-world event that is to be ordered with respect to database entries; this, in turn, selects an event that was used as an anchor of the timestamping sequence; all comparisons between database items and the disputed real world event are replaced by comparisons between database items and the anchor event, and the chronological ordering proceeds as

outlined above.

It is worth noting that one approach a dishonest researcher could use to avoid being identified during an audit process is to add a sufficient amount of noise to data he copies from other researchers. In spite of the fact that the comparison of data (between the accuser and potential wrongdoer) allows for judgments of degrees of similarity, it is clear that a sufficient amount of introduced noise would thwart any meaningful comparison effort. However, a dishonest researcher willing to do this might just as well simply manufacture the entire data set to begin with. The defenses against such abuses are beyond the scope of this paper.

VI. SUMMARY AND FUTURE WORK

We provide the technical foundation to address an important problem in the field of conservation, taking into consideration the social aspects guiding people's behavior. By careful incentive design, we construct a solution that we believe offers a clear advantage over current designs in the realm of conservation databases.

Technically speaking, the novelty of our contribution lies more in the realm of the application we propose and construct than in the actual technical construction. Still, while we draw strongly on known techniques in cryptographic timestamping, we also introduce new constructions relating to verification of claims of when an event occurred. Here, the verification task is complicated by the fact that owners of data are permitted to edit and erase data. Moreover, our holistic security approach—which addresses a combination of practical threats and incentive considerations—also is of potential independent value, as several of its aspects may be applicable in disparate settings.

While our proposed techniques allow for a verification of data ownership, this does not automatically translate into academic recognition of ownership. The problem is not dishonesty, but the fact that insights gained from aggregated data would not automatically be associated with the contributing sets of data. While this is straightforward to achieve in a naive sense, any such approach would result in crediting of both instrumental data and ever so slightly relevant data (the latter which might be a tremendous amount), and with no clear way to distinguish between the two. It would be a worthwhile effort to study and address this problem.

ACKNOWLEDGMENTS

We wish to thank the anonymous reviewers for their helpful feedback.

REFERENCES

- [1] L. von Ahn, M. Blum, N. Hopper, and J. Langford, "CAPTCHA: Using hard AI problems for security," In Proceedings of Eurocrypt, 2003.
- [2] BIOTA The Biodiversity Database Manager, viceroy.eeb.uconn.edu/biota
- [3] J.M. Cherry, C. Adler, C. Ball, S.A. Chervitz, S.S. Dwight, E.T. Hester, Y. Jia, G. Juvik, T. Roe, M. Schroeder, S. Weng, D. Botstein, "SGD: Saccharomyces Genome Database," *Nucleic Acids Res* 1998 26(1), pp. 73-80.
- [4] C. Collberg, S. Kobourov, J. Louie and T. Slattery, "SPlaT: A System for Self-Plagiarism Detection," IADIS International Conference WWW/INTERNET 2003, Algarve, Portugal 5-8 November 2003.
- [5] The Committee on Data for Science and Technology (CODATA), www.codata.org/index.html/
- [6] FIPS 180-2, Secure Hash Standard (SHS), August 2002. csrc.nist.gov/publications/fips/fips180-2
- [7] FIPS 186, Digital Signature Standard, 1994. <http://www.itl.nist.gov/fipspubs/fip186.htm>
- [8] The Global Biodiversity Information Facility (GBIF), www.gbif.org/
- [9] P. Golle, K. Leyton-Brown and I. Mironov, "Incentives for Sharing in Peer-to-Peer Networks," Proc. of the 2001 ACM Conference on Electronic Commerce, 2001.
- [10] S. Haber and W. Stornetta, "How to timestamp a digital document," *Journal of Cryptology*, vol. 3, pp. 99-111, 1991.
- [11] B.S. Harper, C.R. Pyke, H.E. Fox, J.C. Haney, M.A. Schlaepfer, and P. Zaradic, "Gaps and Mismatches between Global conservation Priorities and Spending," *Conservation Biology*, vol. 20, pp. 56-64, 2006.
- [12] M. Jakobsson and S. Myers, "Phishing and Countermeasures: Understanding the Increasing Problem of Electronic Identity Theft," Wiley, ISBN: 0-471-78245-9
- [13] B. Laurie and R. Clayton, " 'Proof-of-Work' Proves Not to Work," The Third Annual Workshop on Economics and Information Security (WEIS04), 2004.
- [14] G.F. Midgley, " Projecting and monitoring climate change impacts on terrestrial biodiversity: Roles for GBIF," GBIF Science Symposium: The role of GBIF and other new technologies in conservation and monitoring biodiversity change, www.gbif.org/GBIF.org/gbif_symposia , 2006.
- [15] C.S. Parr and M.P. Cummings, "Data sharing in ecology and evolution," *Trends in Ecology and Evolution*, vol. 20, pp. 362-363, 2005.
- [16] Proofspace, www.proofspace.com/
- [17] R.L. Rivest, A. Shamir, L.A. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the ACM*, Vol.21, No. 2, 1978, pp. 120-126.
- [18] M. Sharman, " Biodiversity data acquisition and assessment after the MEA: How GBIF and GEOSS will benefit Africa," GBIF Science Symposium: The role of GBIF and other new technologies in conservation and monitoring biodiversity change, www.gbif.org/GBIF.org/gbif_symposia , 2006.
- [19] S. Simiyu, " Implementing the Global Strategy for Plant Conservation in Africa: The role of GBIF," GBIF Science Symposium: The role of GBIF and other new technologies in conservation and monitoring biodiversity change, www.gbif.org/GBIF.org/gbif_symposia , 2006.
- [20] Specify Biodiversity Collection Software, www.specifysoftware.org/Specify
- [21] E.O. Wilson, "On the Future of Conservation Biology," *Conservation Biology*, vol. 14, pp. 1-3, 2000.