PeerSec: Towards Peer Production and Crowdsourcing for Enhanced Security

Zheng Dong
Indiana University

L. Jean Camp
Indiana University

Abstract

Peer production and crowdsourcing have been widely implemented to create various types of goods and services. Although successful examples such as Linux and Wikipedia have been established in other domains, experts have paid little attention to peer-produced systems in computer security beyond collaborative recommender and intrusion detection systems. In this paper we present a new approach for security system design targeting a set of non-technical, self-organized communities. We argue that unlike many current security implementations (which suffer from low rates of adoption), individuals would have greater incentives to participate in a security community characterized by peer production. A specific design framework for peer production and crowdsourcing are introduced. One high-level security scenario (on mitigation of insider threats) is then provided as an example implementation. Defeating the insider threat was chosen as an example implementation because it has been framed as a strictly (and inherently) firm-produced good. We argue that use of peer production and crowdsourcing will increase network security in the aggregate.

1 Introduction

Peer production is considered as an effective solution to generate goods and services. Take Wikipedia as an example. Instead of hiring professionals in different fields, contents of this online encyclopedia are composed by a large number of volunteers. Although contributors do not receive any form of monetary payment, this influential online service has covered articles in far more areas than many professional encyclopedias.

Crowdsourcing is a popular approach for firms that want to outsource the production of goods and services. Crowdsourcing can attract a scale of expertise heretofore not readily accessible to firms. By crowdsourcing products to other entities, a firm can potentially lower the costs of production and increase satisfaction rates among the consumers [1]. Peer production and crowdsourcing have not been systematically leveraged in the design of security systems despite the documentation of the importance of individual home users' decisions. For example, malware infections have significantly increased during recent years [28, 3]. Several defensive options have been made available to online users, but the overall percentage of systems being regularly patched is still low [15]. According to a vulnerability report published by *Qualys* [26], 80% of known security vulnerabilities were found in their scan of more than 40 million IPs; nearly half of IPs had unpatched critical vulnerabilities.

Consider the following example. Everyone would benefit were there no spam. However, even if an individual ensures that her own machine is not part of a botnet, then she still will not see a decrease in spam received. The marginal benefit will be either entirely or at least overwhelmingly to other spam targets in the aggregate. People thus do not feel obligated to invest the time, accept the possibility of lack of compatibility when applying patches, or risk unwanted DRM installations that are a component of patching just to protect some indeterminate strangers from a marginal increase in spam. Incentive aligned design is a chronic problem [2].

Therefore, we propose an innovative approach to enhance the overall investment in security by incorporating peer production and crowdsourcing in security design and implementation. Essentially, we propose designing for communities and social networks instead of relying on the software adoption of individual users. We propose that peer production could to some extent address incentives such that each participating individual values contribution and therefore improves the level of security for an entire community. We organize the paper as follows: Sections 2 and 3 review relevant notions in peer production and crowdsourcing, respectively. Section 4 describes how security systems could be designed for self-organized communities with peer production or crowdsourcing. Section 5 illustrates a high-level instantiation of crowdsourcing for mitigating the insider threat.

Section 6 summarizes our contributions and concludes the paper.

2 Related Work: Peer Production

Instead of production by the firm, peer production relies on participants in self-organizing communities. Contributors to peer production are not necessarily experts in a specific field; in fact, any individual in the general public may participate [12]. Successful products or online services such as Linux, Wikipedia and SourceForge are all examples of peer production. The notion of commons-based peer production or social production [30] was defined by Dr. Yochai Benkler in his work *Coase's Penguin, or Linux and the Nature of the Firm* [4]. While it is clear that not all goods or services may be produced through peer production, Benkler identified the advantages of peer production, which partially lie in the better alignment of available resources with suitable individuals (human capital). It was also documented that *modularity*, *granularity*, and *low cost of integration* are three major requirements of a successful peer production project.

While the creation of many products or services could be *modular* to some extent, there still exist a considerable number of processes that are atomic. Modular processes are those that can be broken into discrete components; each one succeeding or failing independently of others. Atomic processes are those which cannot be broken into individual components but are indivisible entities. Some challenges, such as those posed by creating accounts in a firm may not admit of a peer produced solution. Peer production may, however, be applied to several security measures such as patching individual systems. *Granularity* is another important requirement for peer production. That is, the challenge needs to be of a scale appropriate to apply peer production. As an example, peer-produced solutions such as PGP allows authentication by aggregating individual claims. The *integration cost* could also become a barrier when evaluating the possibility of peer production. For successful open source software developments, documents describing module functionalities and interfaces are highly valued.

Popular social networking websites such as Facebook, Twitter, Flickr, and LinkedIn generate their contents mainly through peer production (personal updates) from account holders. In addition to the functionality of communication, peer production is also commonly utilized in resource recommendations. As Google's '+1' button being widely integrated into millions of webpages, Google Plus users can quickly identify recommended online articles marked by their friends. Furthermore, the top-rated news stories always display on the first page of the social news website Digg[1]. Other reputation-based web services share bookmarks [22], movies [25], and music [9] ratings among online social network members.

Currently, there are a few security designs that incorporate the notion of peer production. For example, Wu et al. have proposed a collaborative intrusion detection scheme in which observations from different detectors are aggregated. With the help of peer production, the accuracy of intrusion detection alerts could be significantly improved [31]. As malicious websites proliferate, Camp et al. have designed the NetTrust system [5] in which individual website blacklists are uploaded and shared among members in several constructed online communities. Participants could benefit from personalized browsing advice from their friends who have visited the same websites before, and receive professional suggestions from a trusted security organization. Moore et al. have considered the wisdom of crowds in detecting phishing sites [23].

3 Related Work: Crowdsourcing

Crowdsourcing is an approach for firms to outsource the production of goods or services to an undefined set of entities [13]. Crowdsourcing is distinct from peer production because while production is distributed there is a single organizing firm. Crowdsourcing has also been utilized by government agencies to complete tasks [10]. Another distinction between peer production and crowdsourcing is that the contributor(s) of crowdsourcing could be an individual, another firm, or a number of people (as in peer production) as noted by Jeff Howe, a key contributor to the notion of crowdsourcing [14].

Firms can lower the cost on content composition by integrating materials submitted by individual participant(s) via crowdsourcing. This yields the further benefit of reflecting greater swathes of public opinion as compared to content created only by firms or the government [1]. There are, however, potential risks created by crowds. For example, the Chevy Tahoe online ad contest [7] is most notable for its humorous satirical advertisements. This crowdsourcing practice has clearly diverged from the original purpose of the firm. Attackers use crowds as well. It has been reported that crowds can coordinate to create malware. Currently spam and malware appear to be both firm produced [17] and quite capable of leveraging social networks [19, 16].

Intuitively, the above description seems to imply that goods and services would always be produced at a lower price when firms choose to contract them out to other suppliers. This viewpoint, however, overlooks other potential costs of outsourcing (e.g. cost for integration). There are several industries (e.g. aeronautics, medical equipments) in which accuracy and provenance are critical. In that case, firms or organizations need to evaluate

[1] http://www.digg.com

the benefit of including crowdsourcing against potential loss from outsourcing a product or service. Economist Ronald H. Coase explained the conditions of the emergence of firms in his work, *The Nature of the Firm* [8]. Additionally, the notion of division of labor was described by Scottish philosopher Adam Smith with the famous example of a pin factory [27]. The strength and benefits of firm-produced security are illustrated in successes and shortcomings of production by firms and crowds respectively. Security conceptions of security as a *private good* with *externalities* [6] has not resulted in adequate public adoption, nor have approaches which consider security as a *public good* [29] or security as a *common-pool resource* [24].

4 Designing for Peers & Crowds

In this section we identify seven steps to build a successful peer-produced or crowdsourced security solution. In Section 5 we describe a more complete instantiation using peer production and crowdsourcing.

1. Identify specific threat leverage points. Thousands of known security threats exist and new threats are reported daily. System design in any case must begin with a clear security goal that addresses one or more security threats. For instance, end users and Internet service providers (ISPs) would take diverse countermeasures to stop spam emails. This is a fundamental condition for a successful system design, since targeting different types of threats may involve distinct assumptions and hypotheses.

2. Make the desired behavior explicit. Once specific security threats and design contexts have been fixed, clear definition of the desired human behavior is the next requirement. For example, *prompt system patching* has been documented to be the most effective way to reduce the number of machines vulnerable to threats [18]. In the scenario of e.g. mitigation of insider threats, the desired behavior would be to avoid access to sensitive information. Our proposed security instantiation in Section 5 promotes *contributing to communities* as another desired behavior.

3. Define parameters of acceptable behavior and reputation. That is, a reputation component needs to be built to indicate ratings of community members based on their previous security-related activities. Reputation could also be used in determining the necessity of recovery. For example, suppose a community member is blocked from accessing the network once ten spam emails are sent from his IP address. In this simplified anti-spam mechanism, *10* is the parameter for reputation and acceptable behavior.

4. Enable communities to self-organize. Self-organization is a crucial component to *peer production*. This organizational structure should not only lower the cost of system maintenance, but also enables security to be achieved more efficiently. Additionally, since a peer production community is not managed or enforced by a firm or an organization, it can potentially create a higher incentive for participants to contribute to their communities.

5. Limit size and number of communities as appropriate. From an economic perspective, determining the correct size of a community is a necessary component for the maximization of social welfare [24]. Intuitively, an individual would have a higher incentive to contribute to a community if he or she knows the rest of the members. Further, while a larger community may mean a larger knowledge base, it may also lead to a higher probability of including a malicious member. Indeed, a small number of participants may already be sufficient to cover the majority of group knowledge; according to a previous study on browsing history of homogeneous undergraduate students, a randomly selected group of 10 students could cover 95% of the distinct websites visited by over 1K students [11]. Self similarity (i.e., homophily) can be leveraged in system design and should at least be considered in step 2. Self similarity occurs on the scale of the individual, that is an individual's behavior on one day is best predicted by *that same* individual's behavior on previous days. Self similarity also occurs in self-selected groups, as individuals tend to join groups with congenerous others.

6. Make community reputation visible. That is, each community member should be able to distinguish a reputable individual from another individual with lower reputation. Intuitively, this visibility would potentially incentivize community members to build their reputations by contributing to communities. Provable exportable reputations are not part of this work.

7. Implement usable interaction to enable mitigation. That is, make clear and simple detection and mitigation actions such that specific solutions are defined for each potential undesirable participant behavior. For instance, once a vulnerable machine is discovered through system scans, it receives and applies a patch. The machine could also be isolated from the rest of the network until the recovery process has completed. In an insider threat scenario, employees could be notified by system warnings and assisted by wizards.

5 New Instantiation of Security

Insider threats have become one of the major sources of cyber-attacks. However, there is also lack of efficient defensive solutions against such threats. Unlike prevention of attacks from outside a network, purely technical approaches have either limited effects or degrade the overall user experience. In this section we briefly review previous research on *budget-based access control*, one of

the proposed solutions to solve the insider threat problem, and then offer an innovative upgrade to this framework based on crowdsourcing. Here we conceive of the risk budget as functioning both as organizational feedback and an employee's reputation.

5.1 Budget-Based Access Control

In previous work, Liu et al. proposed an access control framework for a firm [21]. "Insiders" refer to the employees who have been granted privileges to access some kind of sensitive information (e.g. financial data, human resources data) because of their positions or duties. It is difficult to prevent individuals from abusing the access privileges since a computer application could not automatically distinguish between an ordinary data access and an unnecessary and/or threatening access. With the risk budget, appropriate incentives are provided to each individual with data access privileges so that individual incentives align with organizational interests. Note that incentives are not equivalent to rewards: negative actions (e.g. warning, reduction in wages, demotion) could also be parts of individual incentives. Liu et al. identify the following types of insiders: 1) Inadvertent individuals who do not perform actions that are potentially risky; 2) Careless insiders who perform some risky actions, but not intentionally; 3) Malicious insiders who perform risky actions intentionally and may affect other community members; and 4) Malicious outsiders who have subverted one or more machine(s) of the community to obtain insider status and privileges.

The core component to this access control framework is the risk budget mechanism. That is, potential risks to a firm are measured by *risk points*. At the beginning of a risk budget period, the company calculates its overall risk budget, and allocates it to individuals for whom access to sensitive information are needed. Each data access operation is marked with a *price* in risk points, and results in a deduction of risk point(s) from an individual risk budget. Individuals with a certain risk budget would therefore evaluate the necessity of their actions according, in fact, to organizational risk. Insufficient risk point will lead to refusal of access control requests. However, few exceptions could be made upon an employee's request. By monitoring the individual risk budget, a company could reward an employee with a higher risk point balance; employees with zero or negative balance would be subject to a review of their data access or to some form of punishment.

5.2 The Insider Threat

While previous *budget-based access control* frameworks have been designed for a firm, we propose that the mechanism of a *risk budget* could be migrated to crowdsourcing designs to further improve its efficacy. This upgraded design is not merely an extension of previous research. Instead, we identify different contexts of the threats and build an innovative approach based on self-organized communities. Even with the risk budget mechanism originally introduced in the previous framework, we have significantly modified the process of implementation, enforcement and execution to fit in the new design. The new design does not focus only on a firm, but on a self-selected community of individuals who regularly interact. Any potentially risky action from a security perspective may be considered as threats, especially those affecting community members (e.g. virus-infected email, traveler in despair scams). To design a budget-based access control in a community, the group or the firm may set aggregate risk. That is, the group defines norms of community behavior. This will require some peer production for risk pricing for behaviors or top-down definitions of risk. Notice that the selection of the budget does not need to require the calculus of risk; and a simple interaction can provide a sense of risk-averse to risk-seeking.

Employees may be initially assigned to groups or self-select. One possible approach would be assigning homogeneous individuals to the same group. The advantage of this approach is that each group member will utilize roughly the same percentage of the aggregate budget. The threat of assigned groups is that negative social interactions create perverse incentives. In other words, since homogeneous individuals are not necessarily good friends, group members may have few incentives to contribute to security in the community. Alternatively, communities could be self-organized by their members. That is, group members could invite other individuals in their social networks to join their community.

Generally, each individual with the same tasks will be allocated the same risk budget at the beginning of each risk budget period. The risk budget and point balance for a given individual do not change when an individual joins or leaves a group. Similar to the previous firm-based approach, some quantity of risk points are deducted from an individual's budget each time an action is taken. Exhaustion of an individual's risk budget may lead to warnings, damage to one's reputation, or exclusion from the community. If a *careless insider* or *malicious insider* have been invited to a group in which other individuals are *inadvertent*, there is a strong incentive to exclude that person or *police* his or her potential risky behaviors. By "police" we refer to enforcing social norms [20] or the threat of exclusion from the community.

Another effort in our design to enhance the overall level of security is to make individual and group reputations visible to a wider audience, i.e. a *crowd*. Specifically, the balance of an individual's risk points could be an important indicator of one's former risk-averse be-

haviors. Essentially, we argue that the total risk budget balance and balance of other group members' points be made visible. This approach leverages social norms and a commonly shared desire not to harm one's employer. The previous approach not only promotes compliance to norms by individuals according to policy, but also enables the identification of malicious groups.

Another challenge of this community-based design is to identify a potential malicious individual from other community members while preserving individual privacy. This is solvable with sharing risk point aggregates between groups. This approach requires that each group releases the amount of its budget that has been used to other groups. It is given as a percentage of overall group budget. (Different groups have different baselines.) Imagine that Alice, Bob and Carol are in an *inadvertent* community A (category 1), while Eve is in a *careless* or *malicious* community B (category 2 or 3). Although Eve's point balance is not released to Alice (they are not in the same group), Eve would still be a suspicious person since Alice knows that members in group B have a lower point balance on average. Therefore, even if insiders of categories 2 and 3 may choose to group together, they can still be quickly identified.

For a stronger framework, groups could share aggregate as well as individual risk budgets. So if Alice, Bob and Carol are in one group, when Alice goes over she begins to exhaust the points available to Bob and Carol. This would be appropriate when Bob and Carol have some persuasive power of influence over Alice. For example, in environments where peer evaluation is a component of professional evaluation (e.g., faculty members or Microsoft Research).

To summarize, the roles of the parties are as follows. 1) The organization selects the cost of each risk-creating action, provides identity management for individuals, and sets aggregate goals. 2) The employees implicitly set risk aggregates for employee classification through their longterm risk behaviors; e.g. normal is defined by longterm employee behavior. This is crowdsourced production. 3) The employees in each group limit risk-taking by group members via setting norms, individual or group reputations, or policing through aggregate budgets. This is peer production. We argue our proposed framework could significantly mitigate insider threats. Given appropriate incentives (e.g. risk points), the *inadvertent individuals* would stay benign, *careless insiders* would stop potential risky behaviors when a warning is generated from the reputation system (e.g. deduction of risk points), and the budgets of *malicious insiders* and machines controlled by *malicious outsiders* might decrease more significantly than normal members and could therefore be easily identified. This also potentially resolves the problem of determining potential cooperating parties after a malicious insider is located. The new design also eliminates detailed monitoring by firms or organizations if it is implemented in a firm environment. We do not address enforcement but argue that software and hardware options are available. For example, *Trusted Platform Module (TPM)* may play an enforcement role to prevent administrator tampering. For example, system administrators could be assigned to the same group, in this way that their behaviors can be monitored and policed more effectively. One rough system administrator could be easily detected. (Although a firm with entirely hostile administrators is unlikely to be helped by any design we have developed.) At the least, the sharing of aggregates within a group would make tampering by administrators visible to multiple people, and therefore more likely to be detected.

As future work, we propose three sets of experiments. First we will choose a single function, i.e. access control, and develop the risk pricing to illustrate the feasibility of order-of-magnitude risk pricing. Second, we will examine different communications methods to communicate the risk balance to the individual. We will then test and improve this communication, expanding on the user experiments conducted by Liu et al. [21]. Third, after we have determined that the underlying budget and the interaction are feasible, we will examine the peer production components. Our tentative experimental design begins with two sets of participants. They are divided into groups by *self-joining* and *passive assignment* methods. For each set, we require the participants to complete identified tasks in the lab (i.e., tasks associated with risk points). Individual and group risk budgets are shown to participants as described in our design above. The response of the participants (e.g., whether to complete a task with a risk point reduction or not to complete a task) will reflect the efficacy of this peer-centered design. We hope to follow this with an improved design and longterm experiments of normal use; the first with a student population, the second in a workplace.

6 Conclusion

As malware infections proliferate, computer security measures such as regular system scans, patching, and recovery services continue to be underutilized by end users. In this paper, we propose that security solutions could alternatively be designed for a number of self-organized communities. We explain our design method through detailed instructions concerning the design of a security system based on peer production and crowdsourcing. A high-level instantiation on the mitigation of insider threat has also been described as a proof of our proposed concept. We argue that investment in security can be increased if security solution designs are based on peer-produced communities rather than on hyper-rational

or altruistic individuals.

We enumerate the necessary conditions for system design. We provide a design example, albeit one that must be tested and built in situ for validation.

7 Acknowledgements

We thank John McCurley for his editorial comments. This material is based upon work supported by the National Science Foundation under Grant 0916993. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

[1] ALSEVER, J. What is crowdsourcing? - cbs news. http://www.cbsnews.com/8301-50512_162-51052961/what-is-crowdsourcing/, [Online; accessed 25-March-2012].

[2] ANDERSON, R. Why information security is hard-an economic perspective. In *17th Annual Computer Security Applications Conf.* (Washington, DC, USA, 2001), ACSAC '01, IEEE Computer Society, pp. 358–.

[3] APWG. Phishing activity trends report, first half 2011. http://www.antiphishing.org/reports/apwg_trends_report_h1_2011.pdf, Dec. 2011.

[4] BENKLER, Y. Coase's penguin, or linux and the nature of the firm. *The Yale Law Journal 112*, 3 (2002), 369—446.

[5] CAMP, L. J. Reliable, Usable Signaling to Defeat Masquerade Attacks. In *WEIS: Workshop on the Economics of Information Security* (2006).

[6] CAMP, L. J., AND WOLFRAM, C. Pricing security. In *CERT Information Survivability Workshop* (2000), pp. 31–39.

[7] CNET. Chevy tahoe's online ad contest. http://news.cnet.com/1606-2_3-6056633.html, [Online; accessed 25-March-2012].

[8] COASE, R. H. The Nature of the Firm. *Economica 4*, 16 (Nov. 1937), 386–405.

[9] COHEN, W. W., AND FAN, W. Web-collaborative filtering: recommending music by crawling the web. In *9thIWWW: Int'l Journal of Computer and Telecommunications Networking* (2000), pp. 685–698.

[10] DAVIS, A. Crowdsourcing and democracy. http://blogs.computerworld.com/20101/crowdsourcing_and_democracy, [Online; accessed 2-May-2012].

[11] DONG, Z., AND CAMP, L. J. The decreasing marginal value of evaluation network size. *SIGCAS Comput. Soc. 41*, 1 (Oct. 2011), 23–37.

[12] FARLEX. User-generated content. http://encyclopedia2.thefreedictionary.com/Peer+production, [Online; accessed 25-March-2012].

[13] HOWE, J. The rise of crowdsourcing. *Wired Magazine 14*, 6 (June 2006).

[14] HOWE, J. Crowdsourcing: A definition. http://crowdsourcing.typepad.com/cs/2006/06/crowdsourcing_a.html, [Online; accessed 25-March-2012].

[15] INSPECTOR, S. P. S. Blog post: Your security: 1 in 5 applications are not patched! http://secunia.com/blog/17/, [Online; accessed 25-March-2012].

[16] JAGATIC, T. N., JOHNSON, N. A., JAKOBSSON, M., AND MENCZER, F. Social phishing. *Commun. ACM 50*, 10 (Oct. 2007), 94–100.

[17] KANICH, C., WEAVERY, N., MCCOY, D., HALVORSON, T., KREIBICHY, C., LEVCHENKO, K., PAXSON, V., VOELKER, G. M., AND SAVAGE, S. Show me the money: characterizing spam-advertised revenue. In *20th USENIX Security* (2011), SEC'11.

[18] KELLEY, T., AND CAMP, L. J. Online promiscuity: Prophylactic patching and the spread of computer transmitted infections. In *Workshop on the Economics of Information Security (WEIS'12)* (Berlin, Germany, June 2012).

[19] KLEINBARD, D., AND RICHTMYER, R. U.s. catches 'love' virus. http://money.cnn.com/2000/05/05/technology/loveyou/, [Online; accessed 25-March-2012].

[20] LESSIG, L. Social meaning and social norms. *University of Pennsylvania Law Review 144*, 5 (1996), pp. 2181–2189.

[21] LIU, D., WANG, X., AND CAMP, L. Mitigating inadvertent insider threats with incentives. In *Financial Cryptography and Data Security* (2009), R. Dingledine and P. Golle, Eds., vol. 5628 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp. 1–16.

[22] MARKINES, B., STOILOVA, L., AND MENCZER, F. Bookmark hierarchies and collaborative recommendation. In *21st national Conf. on Artificial intelligence - Volume 2* (2006), AAAI'06, AAAI Press, pp. 1375–1380.

[23] MOORE, T., AND CLAYTON, R. Financial cryptography and data security. Springer-Verlag, Berlin, Heidelberg, 2008, ch. Evaluating the Wisdom of Crowds in Assessing Phishing Websites, pp. 16–30.

[24] OSTROM, E. *Governing the commons: The evolution of institutions for collective action*. Cambridge Univ Pr, 1990.

[25] PARK, S.-T., AND PENNOCK, D. M. Applying collaborative filtering techniques to movie search for better ranking and browsing. In *13th ACM SIGKDD KDD '07* (2007), pp. 550–559.

[26] QUALYS. The laws of vulnerabilities: Six axioms for understanding risk. http://www.qualys.com/docs/Laws-Report.pdf, [Online; accessed 25-March-2012].

[27] SMITH, A. *An Inquiry into the Nature and Causes of the Wealth of Nations*. History of Economic Thought Books. McMaster University Archive for the History of Economic Thought, 1776.

[28] SYMANTEC. Symantec internet security threat report trends for 2010. https://www4.symantec.com/Vrt/offer?_requestid=391138&a_id=81631&, [Online; accessed 25-March-2012].

[29] VARIAN, H. R. *Microeconomic Analysis, Third Edition*, 3rd ed. W. W. Norton & Company, Feb. 1992.

[30] WIKIPEDIA. Commons-based_peer_production. http://en.wikipedia.org/wiki/Commons-based_peer_production, [Online; accessed 25-March-2012].

[31] WU, Y.-S., FOO, B., MEI, Y., AND BAGCHI, S. Collaborative intrusion detection system (cids): a framework for accurate and efficient ids. In *19th Annual Computer Security Applications Conf. (ACSAC '03)* (dec. 2003), pp. 234—244.