Secure Distributed *Human* Computation

Craig Gentry[1], Zulfikar Ramzan[1], and Stuart Stubblebine[2]

[1] DoCoMo Communications Laboratories USA, Inc
{cgentry, ramzan}@docomolabs-usa.com
[2] Stubblebine Research Labs
stuart@stubblebine.com

Abstract. We suggest a general paradigm of using large-scale distributed computation to solve difficult problems, but where humans can act as agents and provide candidate solutions. We are especially motivated by problem classes that appear to be difficult for computers to solve effectively, but are easier for humans; e.g., image analysis, speech recognition, and natural language processing. This paradigm already seems to be employed in several real-world scenarios, but we are unaware of any formal and unified attempt to study it. Nonetheless, this concept spawns interesting research questions in cryptography, algorithm design, human computer interfaces, and programming language / API design, among other fields. There are also interesting implications for Internet commerce and the B24b model. We describe this general research area at a high level and touch upon some preliminary work; a more extensive treatment can be found in [6].

1 Introduction

In Peha's Financial Cryptography 2004 invited talk, he described the Cyphermint PayCash system (see www.cyphermint.com), which allows people without bank accounts or credit cards (a sizeable segment of the U.S. population) to automatically and instantly cash checks, pay bills, or make Internet transactions through publicly-accessible kiosks. Since PayCash offers automated financial transactions and since the system uses (unprotected) kiosks, security is critical; e.g., the kiosk must decide whether a person cashing a check is really the person to whom the check was made out. At first, one might expect that the kiosk uses sophisticated biometric tools, advanced facial recognition algorithms, and the like (which is unsettling since such schemes produce false positives, and can often be outwitted by a clever adversary; e.g., someone can try to hold a photograph up to the camera on the kiosk). However, Cyphermint's solution is very simple: a "human computer" at the back end. The kiosk simply takes a digital picture of the person cashing the check and transmits this picture electronically to a central office, where a human worker compares the kiosk's picture to one that was taken when the person registered with Cyphermint. If both pictures are of the same person, then the human worker authorizes the transaction.

A.S. Patrick and M. Yung (Eds.): FC 2005, LNCS 3570, pp. 328–332, 2005.

In this example, a human assists in solving problems which are easy for humans but still difficult for even the most powerful computers. Many problems fall into this category; e.g., so called "AI-complete" problems which occur in fields such as image analysis, speech recognition, and natural language processing. Motivated by the above example, we put forth the notion of secure distributed *human* computation (DHC). Although DHC might sound far-fetched, several present-day situations exemplify this paradigm:

- **Spam Prevention:** Recognizing that humans can more easily identify junk mail than computers, some spam prevention mechanisms [11][12][13] leverage human votes. Each email recipient presses a button if it receives what it considers to be spam. If enough people vote that a given email is spam, it is flagged as such, and an appropriate action is taken.
- **CAPTCHA Solutions:** Ironically, spammers can hypothetically use DHC to further their goal [1], [2]. Consider free email providers who have incorporated special puzzles, known as CAPTCHAs, that are easily solved by humans, but challenging for computers, during the account creation phase to prevent spammers from automatically creating email accounts; spammers, in turn, can farm these CAPTCHAs out to humans in exchange for access to illicit content.
- **The ESP Game:** In the ESP Game [3], two players are randomly paired over the Internet; they are not permitted to communicate, but both view the same image on their respective web browsers. Each player types in words that describe the image. As soon as both players enter the same word, they get a new image. The goal is to get through fifteen images in $2\frac{1}{2}$ minutes, and the players' scores increase according to various factors. The players get entertainment value and the game organizers now have labels for their images, which is valuable for improving image search.
- **Distributed Proofreaders:** Distributed proofreaders (www.pgdp.net) is a project that aims to eliminate optical character recognition (OCR) errors in Project Gutenberg (www.gutenberg.net) electronic books. A (small) portion of the image file and corresponding text (generated by OCR) is given side-by-side to a human proofreader who, in-turn, fixes remaining errors. By giving the same piece of text to several proofreaders, errors can be reliably eliminated.
- **Other examples:** Open source software development loosely falls into the DHC paradigm; here the difficult problem is not something crisp like image recognition, but instead that computers have a hard time automatically generating source code. As another example, consider Wikis, which are online encyclopedias that are written by Internet users; the writing is distributed in that essentially almost anyone can contribute to the Wiki.

APPLICATIONS TO E-COMMERCE AND B24B. Web sites typically rely on three revenue sources: advertisements, subscription fees, and e-commerce. Earning sustainable revenues from the first two sources is hard (e.g., click-through rates on advertisements are around 0.7% [5], and outside of specific niche industries, few will pay subscription fees for premium Internet content).

However, DHC yields another revenue source: companies who want specific problems solved can farm them out to the hundreds of millions of Internet users.

In exchange for solving the problem, some service or good is provided. We note that DHC payments have several advantages over credit cards. First, solving a human computation problem might be faster than fetching a credit card and entering the billing details. Second, credit card information can be compromised (e.g., if the merchant web server is compromised). Finally, credit card transaction fees are substantial, so cannot be used for low-value content. In a sense, then, human computation can form a new type of online currency or bartering system.

As an example, such a mechanism might be useful on the New York Times web site (www.nytimes.com) which provides free access to the day's news articles, but charges a fee for archived articles. Such a fee (while necessary from a business perspective) might deter users – especially since they can probably (illegally) obtain the article text; e.g., it was posted to a mailing list. However, instead of charging a fee, the New York Times could give the user a human computation problem (e.g., transcribing an audio feed into text). In exchange for solving the problem, the archived article can be provided. This concept extends to other service offerings; e.g., music downloads or long-distance minutes for solutions. DHC may also enable the Business-to-Four-Billion (B24b) model [10] which aims to provide digital services (wireless communication, Internet, etc.) to the world's four-billion poorest people. Individually these people have annual incomes less than $1500 – yet they have large collective buying power. Although the economic feasibility of B24b is still very much an open question, providing services in exchange for solving DHC problems seems like a useful approach, since it depends on an abundance of human resources, while avoiding cash transactions. (On the other hand, since we are talking about *human* computation, there are ethical issues to consider – in particular, as with any human service, we should ensure that the market for human computation is not unduly exploitative.)

Related Fields. DHC is relevant to several research disciplines. With respect to information security, one can superficially view DHC as a type of secure multi-party computation (for a survey see chapter 7 of [7]), since it may involve multiple human computations, but perhaps the differences are more striking than the similarities. First, the parties are human beings instead of computers; second, the parties are themselves not providing actual inputs, but are instead providing candidate answers (which themselves can be construed as inputs into a group decision-making process); third, the "function" to be computed may not always have a clear-cut answer; fourth, the computation may be facilitated by a semi-trusted[1], but computationally "weak" server (i.e., it cannot solve AI-complete problems itself); fifth, we may not always be restricted by privacy concerns, although they are important in a number of motivating applications.

To analyze security, we may consider the case where the adversaries are rational, and use game-theoretic tools. Also, since DHC is a form of currency, we may use cryptographic tools that have been developed in connection with e-cash.

[1] Server trust can be minimized by augmenting a DHC system with a voter and results-verifiable voting protocol [4].

Finally, we remark that some related work on secure distributed computation and CAPTCHAs ([8], [9], [2], [1]) has appeared in cryptographic literature. We are well aware that "security" is less of a cut-and-dried issue in the human computation context than in the cryptographic context, but we view this as an interesting research challenge. Of course, DHC also has interesting implications for algorithm & programming language design, and human-computer interaction.

Early Thoughts. We have used basic tools from probability theory and decision theory in the design and analysis of secure DHC systems. First, our analysis shows, interestingly, that in the presence of certain types of adversaries, standard tools like Bayesian inference are worse than simple approaches like majority vote for combining individual answers. Next, by trying to model candidate utility functions for end users, we find several design principles: we should provide payouts to clients in direct proportion to a rating that measures the accuracy with which they provide answers; we should decrease the rating substantially if a provided answer seems to be incorrect and increase it only slowly for answers that appear correct; and finally, we should take extra measures if a client's payout from cheating is potentially high. We discuss these issues in greater detail in [6].

While our work is preliminary, it seems that secure *human* computing presents a new paradigm that is likely to suggest a rich set of research problems.

Acknowledgements. We thank Naresh Apte, Frank Bossen, Neil Daswani, Minoru Etoh, Ravi Jain, Khosrow Lashkari, Phil Mackenzie, and David Molnar, for helpful discussions and pointing us to various examples of DHC.

References

[1] L. von Ahn, M. Blum and J. Langford. Telling humans and computers apart automatically. *Communications of the ACM*, 47(2):5660, February 2004.

[2] L. von Ahn, M. Blum, N. Hopper and J. Langford. CAPTCHA: Using hard AI problems for security. *Eurocrypt 2003.*

[3] L. von Ahn and L. Dabbish. Labeling Images with a Computer Game. *ACM CHI 2004.* See also `http://www.espgame.org/`

[4] R. Cramer, R. Gennaro and B. Schoenmakers. A Secure and Optimally Efficient Multi-Authority Election Scheme. *EUROCRYPT 97.*

[5] X. Drèze and F. Hussherr. Internet Advertising: Is Anybody Watching? *Journal of Interactive Marketing,* 2003, Vol. 17 (4), 8-23.

[6] C. Gentry, Z. Ramzan, and S. Stubblebine. Secure Distributed Human Computation. *Proc. ACM Conference on Electronic Commerce, 2005.*

[7] O. Goldreich. Foundations of Cryptography – Volume 2. *Cambridge University Press,* 2004.

[8] P. Golle and I. Mironov. Uncheatable Distributed Computations. *RSA Conference, Cryptographers' Track 2001.*

[9] P. Golle and S. Stubblebine. Distributed computing with payout: task assignment for financial- and strong- security. *Financial Cryptography 2001.*

[10] C. K. Prahalad and S. Hart. The Fortune at the Bottom of the Pyramid. *Strategy + Business*, Issue 26, Q1 2000.

[11] Spam Net Web Site. `http://www.cloudmark.com.`

[12] Vipul's Razor Web Site. `http://sourceforge.net/projects/razor.`

[13] F. Zhou, L. Zhuang, B. Zhao, L. Huang, A. D. Joseph, and J. Kubiatowicz. Approximate Object Location and Spam Filtering. *ACM Middleware, 2003.*