

Secure and Lightweight Advertising on the Web

Markus Jakobsson * Philip D. MacKenzie * Julien P. Stern †

Abstract

We consider how to obtain a safe and efficient scheme for web advertising. We introduce to cryptography the market model, a common concept from economics. This corresponds to an assumption of *rational behavior* of protocol participants. Making this assumption allows us to design schemes that are highly efficient in the common case – which is, when participants behave rationally. We demonstrate such a scheme for web advertising, employing the novel concept of *e-coupons*. We prove that our proposed scheme is safe and meets our stringent security requirements.

1 Introduction

The recent development of numerous Internet-based businesses, spanning a range from flowers to CDs, demonstrates the potential of Web-based business in physical merchandise. However, it is clear that the Internet, being an electronic media, is better suited for commercial applications whose nature is an exchange of *information* alone.

Advertising is arguably the type of commercial information exchange that is of the greatest economic importance in the real world. Indeed, advertising is what funds most other forms of information exchange, including radio stations, television stations, cable networks, magazines and newspapers. Although some advertising is simply concerned with name recognition and/or general hype, much advertising is of the more informative kind that truly benefits consumers, such as detailing benefits of certain products and showing price differentials with competing products. It is not an exaggeration to say that the market-based economy and society co-exist with advertising in a symbiotic relationship. Advertising has an annual volume in the United States alone of over 187 billion [oA97], which is larger than the GNP of all but about twenty of the richest countries in the world.

The current trend on the Internet suggests that advertising will remain the major revenue producer for the World Wide Web. However, as advertisers pour onto the Web, establishing an e-commerce niche of their own, it is surprising that many technical problems remain

*Information Sciences Research Center, Bell Laboratories, Murray Hill, NJ 07974. {markus,philmac}@research.bell-labs.com

†UCL Crypto Group, Batiment Maxwell, 3 Place du levant, B-1348 Louvain-la-Neuve, Belgium (stern@dice.ucl.ac.be), and Laboratoire de Recherche en Informatique, Université de Paris-Sud, Batiment 490, F-91405 Orsay Cedex, France (stern@lri.fr). This work was done while the author was visiting Bell Laboratories.

virtually untouched. In the physical world, a multitude of methods have been developed to measure the visibility and degree of success of an ad or commercial. These methods range from measures of the number of viewers/readers and the demographics of these, to methods giving direct feedback to the advertiser. It has been found that many of these methods perform very poorly in an Internet setting, due to lack of trust, lack of reliable metering methods, and a lack of direct feedback.

Currently, the most employed method is based on the number of click-throughs of banners and other ads. More specifically, the merchant (the name we will use for the entity which places the advertisement) counts the number of times his site is accessed from various advertising sites, and pays these a fixed amount for each such access. This is a solution rife with problems. It does not distinguish between successful and unsuccessful visits (where a visit is successful if it generates sales), which makes it economically unpredictable both for the advertiser and the merchant. It creates an incentive for advertisers to trick visitors to visit the merchant, a visit which is likely not to be successful. The method also does not distinguish between a “human visit” and a “computer visit”. This is particularly a problem in the light of the use of traffic anonymizers, which can be used by an ad agency to create a large number of income generating click-throughs, whose originator cannot be established by the merchant. In a setting with very little inherent trust, this is an aggravating factor. The method also is less useful for “planned” visits, as opposed to spur-of-the-moment visits. The method is therefore likely to bring out the worst in ad campaigns, at the expense of truthful consumer information. Finally, it only works for hi-throughput advertising, and does not work well in a situation where advertisements are highly directed and selective, and where purchases are large but few.

We therefore take another approach in this paper, in order to overcome these and other problems. We observe that there is one type of advertising method used in real life which does not have the same inherent problems as the click-through solution. In terms of the trust model, we also make use of a concept that is integral in the physical world, but which has not yet made its debut to the electronic world or that of cryptography.

Starting by looking at the advertising method, we see that the use of *coupons* offers a potential solution to problems posed by advertisers, merchants and users. Coupons are a common real-world method of advertisement, in which a merchant provides users with incentives to buy his products in return for automatic feedback on success ratio, and potentially information regarding demographics as well. Bringing this concept to the web promises to be even more successful than in the physical world, as it allows the merchant to collect very specific information about the ad access, such as the time and context in which it was made, but also allows this to be balanced with user requirements on privacy.

In this paper, we introduce the concept of *e-coupons*. E-coupons can be viewed as the electronic counterpart of coupons, such as those commonly found in mailboxes or newspapers. In order to benefit from an e-coupon, a customer needs to interact with the merchant, which allows the latter to check the validity of the e-coupon and to obtain direct feedback of the impact of his ad campaign. As a result, e-coupons provide a new mechanism for advertising with applications to web metering. It allows a highly efficient and very light-weight implementation, which does not require the distribution of software to users, or the establishment of secret keys for users.

Having briefly addressed the problem of feedback on the success of an advertisement, we will now discuss the trust model and the related charging mechanisms: In this paper, we adopt the *market model* from economics. In terms of pricing of ads, this plainly comes down to a distributed auction in which advertisers are asking to obtain the maximum payment for an ad of a given size, and where the merchants are trying to maximize their benefit in terms of the success of their ads in relationship to the price they have to pay. Given a large enough pool of advertisers and merchants (which the web certainly provides) and reasonable mechanisms for generating feedback (which the use of e-coupons provides), standard economic methods can be used to assess the value of an ad. In addition to the basic cost calculations and pricing mechanisms, the market model takes into account more generic “costs” including public opinion and trustworthiness. Thus, the market model intuitively corresponds to a paradigm which could be called “play fair or lose”, which means that in the long run, parties cannot benefit from cheating if that cheating is detectable. In order to defend against various adversarial strategies, while keeping the scheme very lightweight, we focus on making the common case inexpensive, while providing back-up mechanisms for conflict resolution.

Outline: Section 2 discusses related work, Section 3 precisely defines our model, and Section 4 states the requirements and present the design goals of our scheme. We then give an intuition (Section 5) of our solution, before presenting it in details (Section 6). We list and prove claims on our system in Section 7, and finally study the implementation issues in Section 8.

2 Related work

As already pointed out in the introduction, the currently most employed method, which consists of an advertiser displaying banners and being paid according to the number of clicks on that banner, is unsatisfactory. Its two major problems are that the merchant can deny receiving visits, and that the advertiser can generate fake click-throughs. While it is possible to prevent the merchant from lying on the number of visits he received [RAM98], no solution avoids the second problem in the click-through setting. Furthermore, the solution presented in [RAM98] relies on the presence of Javascript, which is sometimes disabled by the user, for efficiency, security or control reasons.

A naive solution which avoids these problems is to let the advertiser display an ad, and then from the (hopefully) increased sales, try to determine how successful the ad campaign was. Apart from being a difficult estimation to make due to other unrelated influencing factors, it is also not well suited for distributed settings with a multiplicity of ad agencies and ad campaigns. While this method obviously requires the merchant to trust the advertiser, this trust can be diminished by obtaining an estimation of the number of hits a site receives (and possibly checking from time to time that the ad is actually displayed.) In this trust setting, the advertisement problem boils down to a related – and possibly as difficult – problem: web metering. Lately, several metering schemes have been developed to allow, in general, a secure estimation of the number of accesses to a given resource, and in particular to determine the number of visits to a certain site.

Metering schemes, which were introduced by Dwork and Naor [DN92] in order to limit junk mail, and by Franklin and Malkhi [FM97] for the use of advertising, fall into two categories. The first category consists of those schemes in which the visitor performs some medium-hard computation, and sends the results to the advertiser, who saves such results as an indication of the amount of computation performed. Schemes of this type, e.g., [FM97], have the drawback that it is not possible for the merchant to distinguish between computation performed by *visitors* and computation performed by the *advertiser*. Also, it is not possible to distinguish between a situation with two different visitors and one with only one visitor coming twice. Schemes of the second category avoid this problem. These are schemes where the visitor gives the advertiser a transcript which is a function of a secret key the visitor holds. Later, such transcripts are used as an indication of how many visitors a site had. Example of this type is [NP98]. Both of the metering solutions require special software, and possibly also secret keys, to be distributed to all users, which seems to prohibit their use on a large scale. Even if this were not a problem, it is not clear how useful the schemes would be for advertising, as they say nothing about the *quality* of the visit – corresponding to the potential gain for the merchant – they merely give an upper bound of the number of visits.

Coupons on the internet have also been investigated in [KRJM98], but in a very different model with different goals. They focus on *limited distribution* coupons and thus most of their effort is spent on setting up heavy mechanisms to prevent coupon exchange and duplication, including requiring the user to embed personal information into coupons. They also have no notion of an advertiser.

Another model for coupons is described in the New York Times [NYT98], where the idea is to download and print coupons that can be redeemed on an actual visit to a store. This seems to be gaining popularity, but is not as applicable to our work, which is more concerned with on-line e-commerce.

The market model that we introduce allows the use of protocols where incorrect behavior is not prevented on-line, but rather detected in a reasonable amount of time. This is very similar in spirit to off-line e-cash systems [CFN88], and to traitor tracing (e.g., [CFN94].)

We also investigate how to detect that the secret information of some participants have been compromised. This is similar in spirit to the notion of fail-stop signatures [PP97], which allowed the detection of valid forged signatures.

3 Model

Our model consists of a number of interacting participants who behave according to a standard economic market assumption. In brevity, this means that participants behave *rationally* and according to their own benefit. For the purpose of this paper, the model we present focuses on Web-based transactions, although it could be extended to other forms of media.

Participants

We have three main types of participants in our system: merchants, advertisers, and users. They behave as follows:

Merchants The merchants are trying to sell products or services to users, and are buying advertising space on certain Web pages in order to generate sales. The goal of merchants is to maximize the benefit per price ratio for their advertising.

Advertisers The advertisers own certain popular web pages and sell advertising space on these pages to merchants. The goal of advertisers is to maximize the income they receive for this service.

Users The users are browsing the web and possibly buying products and services in response to advertising. In general, they are looking for the best service/merchandise at the lowest price. Their choice may be influenced by the reputations of the advertiser and merchant.

There are also two other participants, playing a lesser role in the protocols:

The certification authority We assume the existence of a certification authority who cannot be corrupted.¹

The judge The judge receives complaints and can authorize legal action when presented with evidence of wrongdoing. He is assumed to be fair.

An additional goal of merchants and advertisers is to gather demographics in order to target their campaign/ad distribution better, and to offer users personalized service if so desired. Those goals should be achievable on a per-user basis, as to maintain user privacy.

Adversary model

We consider two types of adversaries. The first one is a *mobile* adversary [OY91] who can play the role of a polynomial number of the main participants, that is merchants, advertisers and users. The second one is a more powerful adversary who is a *read-all* adversary. The read-all adversary was introduced in [JY97], and has the ability to read all the private information of all the main participants. We wish to obtain a scheme which is secure against a mobile adversary, and which enables to detect a read-all adversary.

Trust model: the Market Assumption

We introduce the use of the market assumption for secure protocol design. The market assumption generally states that each participant in the system attempts to maximize its benefit/cost ratio in every interaction (such as a contract or purchase), where benefits and costs are determined to the best of the participant's ability. In our particular model, this implies the following:

1. A merchant will choose to advertise with an advertiser that generates the optimal sales of service/merchandise with respect to advertising cost.

¹Technically, one can use a proactive signature scheme such as [HJJ⁺97, FGM97, Rab98] to make the corruption of its keys difficult even to a strong adversary.

2. An advertiser will attempt to maximize the income it receives from its advertising space.
3. A user will choose the most appealing services/merchandise according to the best quality and best price.
4. Improper conduct, such as not abiding by signed contracts, or simply providing service of a lesser quality than expected by the buyer of the service, will result in damage of the reputation of the service provider. Legal actions may also be taken.

In terms of the security of a protocol, the market assumption means that we do not require any trust between participants, and that a protocol is secure if we provide ways to *detect and trace* abnormal behavior. In other words, we do not try to detect abuses of the system on-line, so as to keep the common case efficient, but we make sure that any abuse is detected in a reasonable amount of time, and that it can be determined who deviated from the expected behavior. This approach is similar in spirit to off-line e-cash systems, where double-spending is *detected and traced*, as opposed to prevented. However, in our system, the action taken upon detection of an abuse might be as simple as not renewing a business agreement.

4 Requirements and design goals

Requirements

We wish to obtain a scheme where a merchant can advertise his products through the help of advertisers, where accurate information about the success of the advertising can be obtained by both parties, and where there is a high cost for improper conduct. Specifically, the core requirements of the scheme are as follows:

Soundness A merchant should be able to create (binding) ads for any offer on his products.

Merchant Legal Protection It should be infeasible to create (binding) advertising for a given merchant without the consent of that merchant.

Advertiser Legal Protection It should be possible for an advertiser to verify that certain advertising is binding for a given merchant, before the advertiser posts that advertising.

Ad recognition A merchant can verify whether a received ad is binding for him.

Conflict resolution It should be feasible for a designated party (such as a judge) to determine if an ad is binding for a given merchant.

Advertiser Information Protection An advertiser should be able to gather enough information so as to decide whether to renew an advertising contract, and at what price.²

²Note that we are not providing means to *enforce* a contract, but means to find out whether a contract was beneficial or not. This is enough in the market model.

Merchant Information Protection A merchant should be able to gather enough information so as to decide whether to renew an advertising contract, and at what price.

Design goals

We now describe several additional properties that should be taken in account when creating an advertising system.

Corruption detection In case the secret storage area of a merchant is corrupted and another participant manages to create binding advertisement for this merchant, the merchant should be able to detect this.

User Privacy The system should not expose user privacy any more than other (web-based) advertising systems.

Lightweight No special hardware/software should be needed on the user side. The computation and communication costs should be kept as low as possible.

5 Intuition

Intuition of attacks

We briefly discuss a few of the possible attacks on an advertising system. Merchants and advertisers agree on contracts prior to the launch of an ad campaign. A contract fixes the goal and the obligations of both participants. Hence, one of the most obvious attacks against an advertising system seems to be for the merchant or the advertiser either not to respect a contract or to lie about what came out of it. We should note here that it is impossible to *force* participants to follow a contract, as an advertiser could simply decide not to distribute ads. That is why we require that both the merchant and the advertiser obtain meaningful information on the result of a campaign related to a given contract, so as to allow them to be able to form sound business decision onwards.

Another attack would be to try to damage the “public image” of a participant. An advertiser would certainly lose some of his credibility if he distributed bogus advertisements. Similarly, a merchant would get a bad reputation if he were to refuse to honor advertisements presented by users who are convinced the advertisements originated from him. For these reasons, we require that there exist a way to verify that a given advertisement is originating from a given merchant, both prior to distribution and after a possible dispute.

Finally, a merchant also needs to be protected against participants who might try to frame him by creating an advertisement that would be attributed to the merchant.

Intuition of solution

In our model, contrary to the current practice on the web, the price paid to the advertiser is fixed *before* the commencement of a given phase of the advertisement campaign. This, in combination with our scheme providing the merchant with feedback on the quality of the

advertiser allows the employment of the market model, and the solution to the problems other solutions suffer. In our scheme, we have that if the merchant is not satisfied, he can simply stop dealing with this specific advertiser, and given short enough time periods for the contracts, this will encourage the advertiser to play it fair: A design principle of our scheme is *play fair or lose*.

Perhaps the most natural way to provide this feedback is to require that the advertiser gives users some information (the e-coupons), that will be transmitted to the merchant through the users. Note that this approach is certainly the most accurate from the merchant's point of view, as he can *precisely* compute his benefits from any given advertisement, which is the money gained from users buying his products with e-coupons, less the price of advertising.

Naturally, we have to make sure that an e-coupon cannot be created from scratch (as it engages the responsibility of the merchant), and cannot be used with the identity of the advertiser removed. Hence, in our solution, we will only let the merchant create the e-coupons. We insure that only the merchant can create binding e-coupon on his products by simply requiring that he signs each e-coupon. (For other security and efficiency reasons, the merchant maintains a database of all the e-coupons he distributes.) Then, the advertiser will simply need to distribute the e-coupons. We require that the advertiser checks the merchant's signature prior to distribution so that he cannot be framed to distribute false ads. The sole burden of the user is to forward the e-coupon he received from the advertiser to the merchant. We do not require the user to perform any verification, as to keep the common case extremely efficient. Upon reception of an e-coupon from a user, the merchant will check for the origin of the coupon (i.e., will check if he actually produced it). In that case, he will allow the user to access the corresponding offer. Additionally, he will find out the origin of the e-coupon, and will be able to assess the relative quality of his various advertisers.

6 Solution

We now present our solution in detail. We assume the use of a heuristically existentially unforgeable signature scheme such as a Schnorr [Sch91] signature or an RSA [RSA78] signature with an appropriate hash function before signing. We also assume that merchants have obtained certificates on their public-keys from the certificate authority and that the certificate authorities also use a signature scheme of the type mentioned above.

Bidding Advertisers and merchants make offers for services and try to reach agreements. Eventually, they formalize those agreements into contracts. Those contracts define in particular the *fixed* price to be paid by the merchant for an ad campaign, whether it is successful or not. This phase is continuous.

Contract initialization When a merchant and an advertiser have reached an agreement upon a contract, the merchant creates an advertisement of the following form:

$$\mathcal{A} = (txt|advertiser_id)$$

where *txt* is a string describing the merchant's offer along with the terms of use, and *advertiser_id* is a string uniquely identifying the advertiser.

He then generates a signature \mathcal{S} on \mathcal{A} . He creates the e-coupon $\mathcal{C} = (\mathcal{A}, \mathcal{S})$, stores it in his database and sends it to the advertiser.

The advertiser checks the validity of the e-coupon, e.g. that \mathcal{S} is a valid signature on \mathcal{A} . If the signature is invalid, he complains to the merchant and rejects the advertisement. If the signature is valid, he enters the contract execution phase.

Contract execution

E-coupon delivery When a user comes to the advertiser site, the advertiser selects, among a set of possible e-coupons, those he sends to the user.³ Let \mathcal{C} be such an e-coupon.⁴

E-coupon redemption When a user comes to the merchant site specified by the e-coupon \mathcal{C} , he sends \mathcal{C} to the merchant. The merchant checks that the string represented by \mathcal{C} is in his database. If it is, the merchant additionally checks that the e-coupon is used as specified in the *txt* section. In that case, the user can take advantage of the merchant's offer. Otherwise, the merchant notifies the user of the problem. In case the e-coupon is *not* in the merchant database, the merchant enters the corruption detection phase.

Corruption detection The merchant checks the signature of the e-coupon. If this signature is valid, the merchant concludes that his private key has been corrupted and takes the necessary security measures. Otherwise, he simply notifies the user of his refusal to honor this e-coupon.

Contract benefits summarization Each participant assesses his benefits from each contract: merchants count how many users bought their products with e-coupons from each advertiser and compare these benefits to the price paid to each of them; advertisers evaluate, for each merchant, what the ad campaign cost them and compare those values to the money they received at the beginning of the contract. Then, merchants and advertisers are ready to re-enter the bidding phase with more accurate information. Similarly, users continually evaluate the benefits obtained from different advertisers and merchants, much like consumers normally do. This guides the users to choose the best offers and to obtain information from the most reliable advertisers. In particular, if an advertiser frequently distributes worthless or wrongful advertising, users will tend not to use his site anymore.

Conflict resolution If the user did not alter his e-coupon but it is deemed invalid by the merchant, the user checks for the validity of the signature in the e-coupon, and obtains and verifies the certificate of the merchant. If the signature, and the merchant's certificate, are indeed valid, the user has means to prove to the judge that he possesses a valid e-coupon. If the certificate is invalid or does not exist, the user concludes that

³The advertiser may choose whichever strategy he desires to select the e-coupons, possibly basing his choice on information he has about the user, or the user behavior.

⁴The user may check the validity of the origin of \mathcal{C} according to his policy, or if he suspects the e-coupon to be a fake one, but he is not required to.

the merchant is dishonest. If the certificate is valid, but the signature is not, the user concludes that he was cheated by the advertiser who provided him with an invalid e-coupon.

7 Claims

We claim that the system above satisfies soundness, merchant legal protection, advertiser legal protection, ad recognition, corruption detection, conflict resolution, and merchant and advertiser information protection.

Claim 1 *The system satisfies soundness, i.e., the merchant can create a binding e-coupon for any offer.*

This follows directly from the fact that the merchant, using his private key, can produce signatures with respect to his public key.

Claim 2 *The system satisfies merchant legal protection, i.e., assuming that a merchant private key has not been compromised, it is infeasible for someone to create a binding e-coupon for a given merchant without the consent of that merchant.*

This follows from the fact that the merchant uses an existentially unforgeable signature scheme to produce the signature \mathcal{S} part of the e-coupon, and that the certification authority uses a similar signature scheme to certify the participants.

Claim 3 *The system satisfies advertiser legal protection, i.e., an advertiser can be convinced that a given e-coupon is binding for a given merchant.*

Using the merchant public-key, the advertiser can determine the validity of the merchant's signature on an e-coupon, and using the certificate issued by the certification authority, he can check the validity of the merchant's public-key itself.

Claim 4 *The system satisfies ad recognition and corruption detection, i.e., a merchant can verify whether a given e-coupon is binding for him, and can detect binding e-coupon that he has not produced himself.*

Recall that in order to verify that an e-coupon is binding, the merchant simply checks that an e-coupon is in his database. If the signature on the e-coupon is invalid, then it will not be in the merchant database. Now suppose that the merchant is presented with an e-coupon which yields a valid signature with respect to his public key but which is not in his database. As he stores in his database all the e-coupons he produces, then his private key has been compromised with an overwhelming probability (or the signature scheme is not existentially unforgeable.)

Claim 5 *The system satisfies conflict resolution, i.e., a judge can verify whether a given coupon is legally binding for a given merchant.*

This proof is similar to the merchant legal protection proof, that is, the proof of Claim 2.

Claim 6 *The system satisfies merchant and advertiser information protection.*

We will show that while no participant can reliably get all the information about the protocol, each of them can get information which is relevant and sufficient to their economic goals.

Advertisers Advertisers know reliably how many e-coupons they distributed on behalf of each merchant, hence, they can compute their benefit for each contract, which is the difference between the (fixed) price they agreed on in the contract minus the cost incurred by the ad campaign itself (which is essentially function of the number of e-coupon they distributed). Note that, when an e-coupon is distributed only when a user makes a specific action (such as entering specific keywords in a search engine), the advertiser can approximate his benefit continuously, which gives him a good way to know when to stop distributing this e-coupon, depending on the margin he is willing to obtain.

Merchants Merchants know the number of e-coupons from each advertiser that were actually used to buy some of their products. From there, they can compute the benefit (the number of additional products sold less the price of the advertising contract) they obtained from each advertiser.

8 Implementation

We now show that our system is very well suited for a lightweight implementation. In particular, we show that (with some modification to the model) it can be implemented through the use of the cookie mechanism, which is today provided by most browsers and servers.

The main observation is that in the most common case, that is, when no conflict resolution is needed, the only burden on the user is to forward a piece of data from the advertiser to the merchant.

We now discuss the implementation details of each step.

- During the contract initialization phase, the merchant has to sign a string and the advertiser has to verify the signature. This should be done by the means of special purpose software, but has to be done only once for each e-coupon. For an implementation, it is likely that the *txt* field will be divided into several subfields. Typically, it should contain a description of the offer, the conditions of applications, an expiration date, the merchant web address, a special field (usually called the magic number) in order to allow external programs to easily recognize e-coupons, along with some other optional subfields.
- During the e-coupon delivery phase, the advertiser simply sends a string (the e-coupon) to the user. A possible implementation could send the user to a web site in the

merchant's domain where a cookie (the e-coupon) is set. Then the e-coupon will be automatically sent whenever the user visits the merchant's sales site. (Note that in this implementation, the advertiser never sees the coupon.) An external program, called as a plug-in upon reception of a piece of data of the e-coupon type, could also be used.

- During the e-coupon redemption phase, the browser would automatically send the cookie back to the merchant (or the user would send it thanks to the external plug-in). Hence, the user interaction is simple. The programs on the merchant side can therefore present different web pages, if the user sends a coupon or not, and if the coupon is valid or not.

Additionally, users can be provided with an external *coupon manager program* which would enable them (at any time) to browse the e-coupons they received, to recall what they offer, and to directly go to the corresponding merchant web site. This program can also allow them to verify the origin of an e-coupon by checking the merchant's signature.

References

- [CFN88] David Chaum,, Amos Fiat, and Moni Naor. Untraceable electronic cash. In *Advances in Cryptology—CRYPTO 88*, pages 319–327. Springer, 1988. Lecture Notes in Computer Science No. 403.
- [CFN94] Benny Chor, Amos Fiat, and Moni Naor. Tracing traitors. In Yvo G. Desmedt, editor, *Proc. CRYPTO 95*, pages 257–270. Springer, 1994. Lecture Notes in Computer Science No. 839.
- [DN92] Cynthia Dwork and Moni Naor. Pricing via processing or combatting junk mail. In Ernest F. Brickell, editor, *Advances in Cryptology—CRYPTO '92*, volume 740 of *Lecture Notes in Computer Science*, pages 139–147. Springer-Verlag, 1993, 16–20 August 1992.
- [FGMY97] Yair Frankel, Peter Gemmell, Philip D. MacKenzie, and Moti Yung. Proactive RSA. In Burton S. Kaliski Jr., editor, *Advances in Cryptology—CRYPTO '97*, volume 1294 of *Lecture Notes in Computer Science*, pages 440–454. Springer-Verlag, 1997.
- [FM97] Matthew Franklin and Dahlia Malkhi. Auditable metering with lightweight security. In R. Hirschfeld, editor, *Financial Cryptography '97*, volume 138 of *Lecture Notes in Computer Science*, pages 151–160, 1997.
- [HJJ⁺97] Amir Herzberg, Markus Jakobsson, Stanislaw Jarecki, Hugo Krawczyk, and Moti Yung. Proactive public key and signature systems. In *ACM Conference on Computers and Communication Security*, 1997.

- [JY97] Markus Jakobsson and Moti Yung. Applying anti-trust policies to increase trust in a versatile e-money system. In R. Hirschfeld, editor, *Financial Cryptography '97*, volume 138 of *Lecture Notes in Computer Science*, pages 217–238. Springer-Verlag, 1997.
- [KRJM98] Manoj Kumar, Anand Rangachari, Anant Jhingram, and Rakesh Mohan. Sales promotions on the internet. In *Proceedings of Third Usenix Workshop on Electronic Commerce.*, pages 167–176. USENIX, 1998.
- [NP98] Moni Naor and Benny Pinkas. Secure and efficient metering. In Kaisa Nyberg, editor, *Advances in Cryptology—EUROCRYPT '98*, volume 1403 of *Lecture Notes in Computer Science*, pages 576–590. Springer-Verlag, 1998.
- [NYT98] THE NEW YORK TIMES. Is coupon clicking the next advertising trend? <http://www.nytimes.com/library/tech/98/09/cyber/articles/13advertising.html>
- [oA97] Newspaper Association of America. All media ad volume, 1997. <http://www.naa.org/marketscope/databank/index.html>.
- [OEC] OECD. OECD news release 1996. <http://www.oecd.org/dac/htm/opodoc.htm>.
- [OY91] Rafail Ostrovsky and Moti Yung. How to withstand mobile virus attacks (extended abstract). In *Proceedings of the Tenth Annual ACM Symposium on Principles of Distributed Computing*, pages 51–59, Montreal, Quebec, Canada, 19–21 August 1991.
- [PP97] Torben Pryds Pedersen and Birgit Pfitzmann. Fail-stop signatures. *SIAM Journal on Computing*, 26(2):291–330, April 1997.
- [Rab98] Tal Rabin. A simplified approach to threshold and proactive RSA. In Hugo Krawczyk, editor, *Advances in Cryptology—CRYPTO '98*, volume 1462 of *Lecture Notes in Computer Science*, pages 89–104. Springer-Verlag, 1998.
- [RAM98] Micheal K. Reiter, Vinod Anupam, and Alain Mayer. Detecting hit shaving in click-through payment schemes. In *Proceedings of Third Usenix Workshop on Electronic Commerce.*, pages 155–166. USENIX, 1998.
- [RSA78] Ronald L. Rivest, Adi Shamir, and Leonard M. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978.
- [Sch91] Claus P. Schnorr. Efficient signature generation by smart cards. *Journal of Cryptology*, 4:161–174, 1991.