USING THE INTERNET TO REDUCE SOFTWARE PIRACY — On Anonymous Receipts, Anonymous ID Cards, and Anonymous Vouchers *

Ralf C. Hauser <hauser@acm.org>

Abstract

Copyright piracy occurs both on the producer side of copyrighted information as well as the consumer side. The producer side experiences two forms of pirating: i) Entirely illegal reproduction and distribution of pirated software, and the more subtle problem of ii) authorized vendors defrauding their software producers, for example, by placing bootlegged copies on their shelves that are indistinguishable by a consumer from authentic packages. This paper discusses how the problems on the producer's side could be alleviated by a different software vending method supported by Internet technologies.

The primary assumption of this proposal is that honest consumers do not want to risk prosecution, because, unknown to them, software sold or installed on their hardware by offending vendors infringes the terms of licensing contracts.

1 Introduction

Instructions in a Microsoft antipiracy brochure [1] illustrate that, with the increased sophistication of pirates, not only the effects of protection mechanisms decrease, but also the ability for honest consumers to recognize illicit merchandise becomes more difficult. There is a lack of inexpensive and effective mechanisms for the honest consumer to determine the authenticity of merchandise.

The goal of this paper is to present a concept concerning how copyrights pertaining to digital material can be managed globally in such a way that honest customers can prove that they acquired material in compliance with the pertinent license conditions. The absence of such proof of honesty may also be recognized in the long run by prosecution agencies as a strong indication of abuse. In the short run, it is hoped that it will become a standard of good practice in the industry to follow the concepts proposed here.

This paper begins by introducing a new software vending method. Thereby conditionally anonymous receipts, ID cards, and vouchers are defined. Subsequently, it is shown how the concept can be extended to share-ware distribution methods and how the trust in this new vending concept can be bootstrapped.

2 Background and Assumptions

2.1 Technologies Employed

The basic technologies for the proposals below are:

- **Secure one-way functions**: Such a function has the property that the input is hashed to a value in such a way that it is cryptographically hard to derive the input from the output. Furthermore, it is important that it also be very difficult to find multiple inputs that lead to the same output. A frequent example is MD5 [2].
- A**symmetric encryption**: $E_y(X) = Z$ is the notation for the encryption of plaintext X by the public key of Y, $D_y(Z)$ is the decryption of cryptogram Z with the private key D_y, which again yields X. RSA is [3] the best known example.
- **Signature**: These algorithms used for asymmetric encryption can also be employed to sign messages in a way that cannot be repudiated. Certification hierarchies may establish global trust relations.
- **Anonymity**: With increased connectivity, the protection of privacy becomes more important. Anonymity can rely on trusted third parties or on cryptographic mechanisms. Mixes [4] are an important concept to provide sender anonymity in packet routing networks.
- **Electronic payment**: Secure, anonymous off-line payment protocols have been a research topic for years. The concept of "wallets with observers" [5, 6, 7] has brought such schemes within the reach of broad applicability - they now also prevent instead of merely detect double spending.

2.2 Starting Position

- The consumer of the software as well as the vendor own at least a PC and have an asymmetric cryptosystem at their disposal, for example in the form of PGP [8] or PEM [9].
- The copyright owners have their public key certified. Optionally, vendors do too (see chapt. 7, TIOS).
- The consumer is capable of obtaining a certified key through the network. Store and forward connectivity of mail is sufficient.
- The software producers/copyright owners add to each product they ship a signed policy document stating the license terms and the payment modalities.

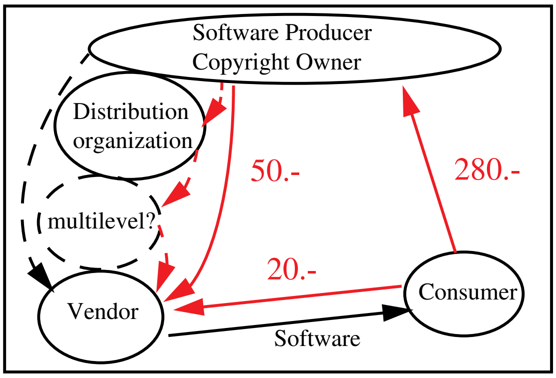
The method protects the producers of software (i.e. the copyright owner) as well as consumers during the distribution of software. If consumers become dishonest after this acquisition phase, nothing in the method hinders them from using the software more intensively than permitted or turning into a fraudulent redistributor.

3 New Software Vending Method

The vendor of a software package must only charge for the direct costs of the distribution medium and manuals etc. The reimbursement of the copyright owner only takes place after the store-sale according to the consumer's trustworthy interpretation of policy and pricing information that is in electronic form.

The financial flows in current distribution systems could be considered to almost encourage fraud. Altering these financial flows may significantly alter the situation, as the following example shows. The method assumes a distribution process with at least three parties, the software producer who owns the copyrights, a vendor, and a consumer:

Figure 1: New Software Vending Method



The non-electronic sale of a commercial software package with a value of, say, $300 could proceed as follows:

1. At the vendor's cashier with a shrink-wrapped package under his arm, the consumer pays $20 and is given a bill for the remaining $280 payable directly to the software producer.
2. This bill is marked with a unique transaction number.
3. The consumer has to sign that he will pay this amount within 30 days and leave his address, for which identification such as a driver's license must be shown.
4. At home, the consumer loads the software onto his computer, obtains the software producer's certificate through a trustworthy channel and verifies the signed policy file, which is also provided on the distribution disk.
5. The consumer compares the amount and recipient of the payment in the bill received from the vendor with the information in the policy file. When all information is correct, he pays the bill.
6. The software producer receiving the payment derives the vendor from the transaction number.
7. With the next delivery of software packages, the vendor receives the outstanding $50 that will complete its sales and profit margin.
8. The vendor also obtains the transaction number and therefore can erase the consumer's address from the local database of outstanding payments.

This method makes it unattractive for vendors to copy software illicitly or to intersperse bootlegged copies on their shelves. If some consumers do not pay in time, the vendor can send them a reminder. In the conventional paper world, there are no anonymous long-distance payments, but essentially, the software producer does not need to know the consumer. Because it is impossible to subvert the signature scheme of the policy files, pirates are unable to divert the money. If the software producer tries to defraud the vendor, the vendor will, once the payment period has expired, remind the consumers, and the consumers can prove the payment by their receipts. Remaining problems can then be solved between the vendor and the software producer out-of-band, possibly with recourse to legal authorities. If the vendor's price was false or the policy under which the software should run was incorrectly announced, the problem is the same as if a customer opens a shrink-wrapped package and the disk is not readable - the shop, hopefully, will still exist and be obliged to take the package back.

If consumers have anonymous electronic payment mechanisms at their disposal or pay through a trustee, they can obtain full anonymity towards the software producer as described next.

4 Anonymous Receipts

The consumer may take an identifier of the product they bought (SW-ID), a random value, their user ID, and optionally further information such as the license policy under which they intend to use the product. This input is entered into a secure one-way function and yields a number Y. This number Y, the Software ID, the chosen license policy as well as the money can then be passed to the paying trustee.

The trustee pays out of its general funds and simultaneously forwards Y and the other information it obtained from the consumer.

The software producer verifies whether the amount of money corresponds to what is due according to the product and chosen license policy. It then writes a receipt containing the date, Y, the amount, product and license information, etc. This receipt is signed with the producer's certified signature key and returned to the payment trustee.

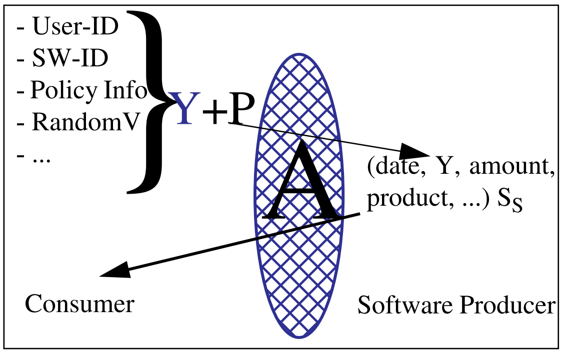
The payment trustee forwards this receipt, and as soon as the consumer has acknowledged the receipt of this anonymous receipt, the trustee must discard all information pertinent to this transaction. Before, the trustee must keep the information in order to be able to retransmit it in the case of communication failures or disputes with the software producer.

The random value in the calculation of Y exists for two purposes:

1. It prevents the software producer from trying to determine the user ID by a brute force attack by iterating through a limited set of possible user IDs.

2. As a variant, any user identification can be omitted in the generation of Y. Revealing the random value is then the "one-show" proof that the holder of Y is indeed the legitimate owner of the receipt.

Figure 2: Anonymous Receipt



S_S = signature key of software producer
Y = hash of the receipt information by a secure one-way function
A = anonymity providing network routing
P = payment mechanism information

The function of the anonymizing trustee can be replaced in the future when the mentioned electronic payment mechanisms are fully available. These mechanisms assume, for example, that the exchanging parties meet at some place and that their electronic payment devices execute the payment protocols by short-distance communications such as by infrared technology. The location therefore does not become a threat to the anonymity of either party. In the scenario of this paper, the payment to the copyright owner is executed remotely and thus the network return address for the receipts may become a threat to the consumer's anonymity. It is therefore advised that electronic payment mechanisms be used only in combination with the aforementioned infrastructures providing sender anonymity on the network level.

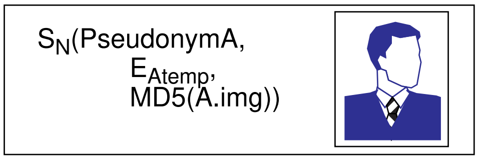
Evaluation:

- Given that the vendor and the trustee do not cooperate with the software producer, this message exchange provides unconditional anonymity between the consumer and the software producer even though it relies on a digital pseudonym, namely Y. The reason for this is that nobody else has knowledge of the real ID behind this pseudonym unless there is a dispute and the consumer has to unveil the components leading to Y to prove his or her ownership of the receipt.
- The method is generally applicable for any product, not only for software.

5 Anonymous ID Cards

There is still no anonymity between the vendor and the consumer in this method. This could be achieved if both the vendor and the consumer trust a notary service and have a certificate to verify signatures from this notary:

Figure 3: Anonymous ID Card



Consumer A identifies her- or himself fully to the notary N. The consumer furthermore creates a new, temporary asymmetric key pair. The notary service registers the consumer's full address and signs the public key without adding any address and name information. A secure hash value of a digital passport picture of the consumer is also included in the signed notary statement, which is called an "anonymous identity card".

Step 3) of the previous sales protocol is changed as follows: The consumer hands a diskette containing this notary statement and the passport picture to the vendor's cashier. The cashier's PC displays the picture of the customer to verify the holder of this digital ID. If this verification succeeds, the ID is stored at the vendor together with the unique transaction ID. If the consumer does not pay in time, the vendor sends a reminder to the notary. Ultimately, the identity of the consumer could be revealed upon entering a legal dispute. This method provides full anonymity for the consumer based on full trust in the notary.

The most sensitive part of the protocol is that the consumer could repudiate having bought the software and claim that the vendor has simply reused an old notary statement to create a reminder. Assuming that a hand-written signature does not destroy the anonymity, the consumer might still have to sign by hand her or his obligation to pay. Then, the problem is reduced to the problem of the original protocol with a consumer repudiating that she or he had given the address and a hand-signature for a specific sale. If the consumer can be expected to carry a device such as a Personal Digital Assistant (PDA), this PDA could cryptographically sign a message specifying the date, the amount, and the transaction ID of the sale employing the "anonymous" key pair given in the anonymous ID.

Fully Networked Approach

If the scenario is to take place in a fully networked environment where the customer no longer walks up to a physical store, the 30-day payment period and the vendor's local database can be omitted. The consumer still presents an anonymous ID to the vendor and the public key in it is used to secure the delivery of the software. The vendor only releases the software when it has received the acknowledgment that the payment has arrived at the copyright owner. In a fully networked scenario, there is also much less reason to delegate the software distribution to an independent agent, i.e., vendors distinct from the producer and the passport pictures in the ID cards become obsolete.

6 Anonymous Vouchers and

Extensions for Share-ware

6.1. Anonymous Vouchers

The mechanism of anonymous receipts can also be used to create anonymous vouchers. To preserve anonymity, the “one-show” version of the receipts without the user ID is taken. A customer can pay anything in advance and, by revealing the random value, the voucher is cashed. This approach requires the vendor/software-producing organization to make sure that the voucher is not cashed more than once. Therefore, they must store a log of vouchers cashed in the past. To keep this task tractable, two measures can be taken:

1. The voucher contains an expiration added by the software producer.
2. The prepaying customer and the vouching producer agree on a limited number of vendor outlets in advance. Adding a vendor ID to the voucher makes it unnecessary to consult a global database to determine the validity of a voucher.

6.2. Extending the method for Share-ware Distribution

Under the assumptions of the model, a software producer cannot prevent the consumer from disclosing the software to additional machine owners. However, having all the described infrastructure and organizational procedures in place, this propensity of the consumers to redistribute could be regarded as an asset instead.

The software producer could explicitly permit the redistribution of the software in the following way:

1. The recipient of such a share-ware-like software distribution among consumers must obtain a signed message from the redistributor specifying the date and product redistributed.
2. The recipients are then obliged to store this message with the software until they have also verified the policy and obtained an anonymous receipt. The recipients have to pay the full amount of $300 to the software producer within 30 days.
3. The redistributing consumers may chose whether they do not want to be known to the software producer or whether the recipient consumers should transmit their bank account ID to the copyright owner in the process of payment. Upon receipt of the money, the software producer sends the manuals and the original diskette to the recipients. In the latter case, the software producer will pay, say, $30, i.e. less than the vendor's profit margin, to the redistributing consumer.

Evaluation

Stating this right of the redistributor to receive $30 also provides an incentive in the bilateral relation of the redistributor and the recipient to obey the licensing terms: Either the redistributor renounces the $30 or she or he obtains it from the recipient directly. If the redistributor obtains the money directly, her or his legal situation will be worse than that of a pirate today. The illicit copy is no longer a peccadillo but the redistributor even makes money off it. The same applies for the recipient: he or she not only obtains illicit software by convenience, he or she actually pays for it in full awareness of the illegal situation. Furthermore, unless they renounce the bilateral transfer of $30, it also involves two active pirates operating jointly.

The main design goal of this approach is still to keep the honest consumer honest, this extension, however, shows that it is even possible to provide additional convenience simultaneously.

The main problem of this extension lies in its enforcement. Recipient consumers will not be reminded to pay unless the redistributing consumer does so. With the enactment of increasingly effective copyright laws, the authorities in several countries have begun to inspect consumer sites for their compliance with license agreements. The signed message from the redistributor or the anonymous receipt may become important evidence in this process. If recipients are inspected and have not yet paid, the message from the re-distributor determines whether they are still in the “contractual time” limit. Within this time of, say, 30 days, the recipients can consider the package a demonstration version and they could still remove it entirely without entering any financial obligation.

This idea of a demonstration version relies on the assumption that it is unlikely that the redistributors would provide the recipients with signed messages covering the entire potential usage time, and the recipients would then carefully manage these messages such that a currently valid message is always visible.

7 Bootstrapping Trust (TIOS)

Essential to all the mechanisms presented is that the honest consumer is capable of obtaining correct information enabling trustworthy verification of the software producer's signature and policy data. The proposal of this paper is therefore to use a Trusted Information Origin Server (TIOS) as a trusted third party of the copyright industry for this purpose. If every copyright owner registers him- or herself and the products with such a TIOS, it may become a requirement of standards of good business practice for system administrators to consult such a server before installing information obtained through the net. The TIOS will produce a key certificate for every registered software producer and have a database about which product was built by which producer much as the US Copyright Clearing Center did. Especially, if more rigid copyright enforcement as outlined above became reality, there would have to be a service enabling common consumers to ensure their own copyright compliance: They either would retrieve the software producer's certificate from the TIOS to verify the signed policy files coming with packages or, in the case where no policy file is present, they would consult the TIOS about the license status of a package by submitting some package characteristic such as a trusted integrity checksum to find out whether the policy file only accidentally got lost. Such a service must have the following characteristics:

- **Security**:
  o The honest consumer must have the assurance that it is impossible for the provider of bootlegged software to subvert the trust-establishing TIOS interactions. This requirement applies both to the identification of software and the software producer as well as to the content of the policy. The consumer is therefore obliged to obtain the public key of the TIOS over a secure, i.e. possibly out-of-band, channel.
  o The relation between the TIOS operating institution and the software producers must be relatively tight and might involve intensive identity checks and the deposit of some guaranteed amount of money (see also [11]). Otherwise, a pirate could register some bootlegged copyright-protected software under a new name and already have disappeared with accumulated "royalties" by the time the plagiarism would be detected.
- **Anonymous**: The TIOS must not trace the consumers consulting its database; otherwise, it may become a disincentive for consumers to employ the service if, shortly after consulting the server, the police raids the consumer to check for illicitly installed software. Therefore the TIOS must also serve requests originating from a service that provides sender anonymity.
- **Inexpensive**: The honest consumer can reasonably be expected to consult a TIOS only if this action entails negligible cost. This is also true for small developers registering a TIOS. Such a service therefore has either to be run by a government organization, some industry-wide software producer association or a consumer organization.

Such a TIOS should be combined with some integrity assurance service such as that provided by Rubin's Betsi service [10].

8 Conclusion

This paper shows how piracy on the vendor or producer side can be reduced without unnecessarily undermine the privacy of consumers. Starting from a concrete scenario of a software sale, the anonymous receipts, ID cards, and vouchers can be employed in much more general scenarios. The paper has furthermore shown how these software sales can also be performed in a share-ware-like way - although assuming the relatively high trustworthiness of at least one of the consumers involved. Last, the design of a trusted information origin server has been sketched. This service enables participants in global networks to comply fully with licensing terms without restricting current distribution methods which may appear anarchic. If implementing the proposed software vending methods and consulting the TIOS becomes part of widely followed standards of good practice, the Internet may avoid becoming subject to rules such as government-imposed, rigid, compulsory license agreements, etc. that may be cumbersome to follow and therefore detrimental to the future evolution of the network.

References

[1] Microsoft. Microsoft international licensing policies: Answers to frequently asked questions, 1994. Dt.: Informationen zur Software Piraterie.

[2] Ronald Rivest. The MD5 message-digest algorithm, April 1992.

[3] Ron L. Rivest, A. Shamir, and Leonard M. Adleman. A method for obtaining digital signatures and public-key cryptosystems. Journal of the ACM, 21(2):120-126, February 1978.

[4] David L. Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. Communications of the ACM, 24(2):84-88, February 1981.

[5] David Chaum and Torben Pryds Pedersen. Wallet databases with observers. In Advances in Cryptology - CRYPTO '92, volume 740 of Lecture Notes in Computer Science, pages 89-105. Springer-Verlag, Berlin Germany, 1993.

[6] Stefan Brands. Untraceable off-line cash in wallet with observers. In Stinson [12], pages 302-318.

[7] Niels Ferguson. Extensions of single term coins. In Stinson [12], pages 292-301.

[8] Philip Zimmerman. The Official PGP User's Guide. prz@acm.org, 1994. The MIT Press. More in http://www.pegasus.esprit.ec.org/people/arne/pgp.html.

[9] D. F. Hadj Sadok and J. Kelner. Privacy enhanced mail design and implementation. COMPUTER COMMUNICATION REVIEW - A Quarterly Publication of the ACM SIGCOMM, 24(3):38-46, July 1994.

[10] Aviel D. Rubin. Trusted distribution of software over the Internet. Technical report, Bellcore, 1994. More in http://www.bellcore.com/SECURITY/security.html.

[11] Charlie Lai, Gennady Medvinsky, and B. Clifford Neuman. Endorsements, licensing, and insurance for distributed services. In Jacques Stern, editor, 2nd ACM Conference on Computer and Communications Security, Fairfax, Virginia, November 1994.

[12] Douglas R. Stinson, editor. Advances in Cryptology - CRYPTO '93, volume 773 of

Lecture Notes in Computer Science. Springer-Verlag, Berlin Germany, 1993.

Author Information

Ralf C. Hauser Ralf Hauser will graduate as a Ph.D. from the Department of Computer Science at the University of Zurich in May 95. He also holds a M.Sc. from the University of Toronto. His research interests are in the field of security, distributed systems, and networked information systems. Currently, he is working with the
IBM Research Division, Zurich Research Laboratory, Saumerstr. 4, 8803 Ruschlikon, Switzerland, Phone:+41/1/724-8426 Fax:+41/1/710-3608

*) Most of this work has been funded as part of the author's doctoral research by the University of Zurich.