Privacy Engineering for Digital Rights Management Systems

Joan Feigenbaum[1]*, Michael J. Freedman[2]**, Tomas Sander[3], Adam Shostack[4]

[1] Computer Science Dept., Yale University, PO Box 208285, New Haven, CT 06520 USA
email: joan.feigenbaum@yale.edu
[2] MIT Lab for Computer Science, 200 Technology Square, Cambridge, MA 02139 USA
email: mfreed@lcs.mit.edu
[3] InterTrust STAR Lab, 4750 Patrick Henry Drive, Santa Clara, CA 95054 USA
email: sander@intertrust.com
[4] Zero-Knowledge Labs, 888 Maisonneuve East, Montreal, Quebec H2L 4S8 Canada
email: adam@zeroknowledge.com

1 Introduction

Internet-based distribution of mass-market content provides great opportunities for producers, distributors, and consumers, but it may seriously threaten users' privacy. Some of the paths to loss of privacy are quite familiar (*e.g.*, mining of credit-card data), but some are new or much more serious than they were in earlier distribution regimes. We examine the contributions that digital-rights-management (DRM) technology can make to both compromising and protecting users' privacy. We argue that the privacy-enhancing technology (*e.g.*, encryption, anonymity, and pseudonymity) that absorbs most of the attention of the security R&D community cannot by itself solve the privacy problems raised by DRM, although it can play a role in various solutions. Finally, we provide a list of "privacy engineering" principles for DRM systems, some of which are easy to implement and potentially quite effective.

The goal of DRM technology is distribution of digital content in a manner that protects the rights of all parties involved, including (but not necessarily limited to) copyright owners, distributors, and users. Appendix A below contains a detailed description of a generic DRM system and its technical components. Here, we give a high-level description that suffices for our purpose, which is exploration of the interplay among DRM systems, user privacy, and business and legal issues.

The following is a list, adapted from Chapter 5 of [25], of some of the security technologies that often play a role in DRM:

– *Security and integrity features of computer operating systems* include, for example, the traditional file-access privileges enforced by the system.
– *Rights-management languages* express in machine-readable form the rights and responsibilities of owners, distributors, and users, enabling application programs to determine whether requested uses should be allowed.
– *Encryption* scrambles digital content so that it can be transmitted or stored in an unusable form (and later decrypted and used, presumably by an application that is authorized to do so and has legitimate possession of the decryption key).
– *Digital signatures* provide assured provenance of digital content and nonrepudiation of transactions.
– *Fingerprinting and other "marking" technology* embeds (usually imperceptible) ownership or rights information in digital content so as to facilitate tracking of copying, distribution, or usage.

DRM-system development consists of putting these pieces together into an end-to-end system that serves the needs of owners, distributors, users, and all other major stakeholders. For high-quality, popular content, this is a daunting development task, and we are not claiming that it is a solved (or even solvable) problem. Nor are we claiming that DRM technology alone can protect everyone's interests perfectly; laws, social norms, and (most importantly) business models will play major roles in making mass-market distribution work, and it will probably never work perfectly. Our claim is simply that the use of these techniques can

* Supported in part by ONR grants N00014-01-1-0795 and N00014-01-1-0447 and NSF grant CCR-0105337.
** This work was largely done while the author was visiting InterTrust STAR Lab.

affect user privacy and that this fact should be taken into account at every stage of DRM-system design, development, and deployment.

At the risk of stating the obvious, we note that there can be inherent tension between the copyright-enforcement goals of owners and distributors who deploy DRM systems and the privacy goals of users. Rights enforcement may be facilitated by user tracking or by network control of users' computers, but both of these are potentially destructive of user privacy. One of the major ways in which privacy can be compromised is through data collection by distributors and network operators. We discuss typical DRM loci of collection in more detail in Section 2 below.

Privacy engineering for DRM is made significantly more complicated by the fact that there are legitimate reasons for distributors and network operators to collect data about users and their activities. These include traffic modeling for infrastructure planning and QoS enhancement; risk management; backup and archiving; targeted marketing and recommendations; mining of aggregated, depersonalized datasets for trend-spotting and (untargeted) marketing and advertising; and counting or statistical sampling (*e.g.*, by clearinghouses like ASCAP/BMI for payments to artists). However, data collected for legitimate reasons can also be used in illegitimate (or perhaps just distasteful or annoying) ways. Furthermore, in today's dynamic business environment, mergers, acquisitions, bankruptcies, and other changes over the life-cycle of a corporation can radically change who has access to what information and how that information may be used and cross-referenced. The merger of DoubleClick and Abacus Direct exemplified how changing access to data can have radical privacy implications [5].

It is our thesis that sound privacy engineering of content-distribution and DRM systems can help defuse at least some of this inherent tension. If properly designed, implemented, and used, DRM can provide reasonable user-privacy protection[1] and simultaneously supply businesses with information necessary for their basic functionality at a fair cost. We give a list of privacy-engineering principles in Section 4 below; perhaps the most important overarching theme of these principles is that data-collection procedures should be tailored very precisely for the business purposes they serve and that personally identifying information (*e.g.*, names and addresses) should be omitted from all data collections whose purposes don't require it.

We focus our discussion on mass-market content. For niche-market content, we believe that privacy issues are less pressing or at least are more easily solved by technology, legal contracts, or some combination of the two.

The rest of this paper is organized as follows. In Section 2, we review the ways in which DRM can impact user privacy. In Section 3, we explain why this problem cannot be solved straightforwardly by the considerable number of tools that have already been explored at length in the security and privacy literature. In Section 4, we suggest a practical approach to privacy engineering using the Fair Information Principles and privacy audits. We provide several *simple, yet effective* principles that engineers should observe for privacy-friendly system design. We conclude in Section 5.

2 How digital distribution and DRM affect user privacy

In this section, we review how Internet content distribution and DRM can lead to privacy loss. We also point out which of these paths to privacy loss are common to all Internet-based commerce or to all forms of mass-market content distribution and which are specifically created or exacerbated by DRM. In order to do this, we must examine some of the ways in which commercial distributors might use DRM to build profitable businesses.

First, consider a straightforward distribution model that does not involve DRM. The user downloads a digital work from the distributor's website; this transaction may or may not involve payment, and, if it does, the user will provide a credit-card number or some other information that allows the payment to work its way through the financial system. Once the user has the digital content on his computer, he is subsequently technologically untethered to the distributor and may use the content in any way that he chooses (albeit at his legal risk if he violates copyright or other laws). There are two main ways in which this type of transaction can pose a threat to the user's privacy. His Web activity can be monitored (*e.g.*, through

[1] In this paper, we use the term "privacy" to mean end-user privacy, *i.e.*, privacy of the individual consumer. We acknowledge that other constituencies, *e.g.*, content owners and technology providers, have privacy concerns, but those concerns are the subject of another paper.

client-side cookies, server-side logs, or a variety of other means), and his credit-card or other payment data can be mined. The first threat is common to all Web-based commerce (and to non-commercial Web activity, for that matter) and has nothing to do with content distribution specifically. The second is a long-standing concomitant of non-cash (or, more precisely, non-anonymous) payment systems, and it has nothing to do with content distribution or with Web-based transactions, except insofar as the migration of commerce to the Web could greatly increase the extent to which payment data are mined. Many more parties can collect and access payment data online than can do so offline.

Our concern in this paper is the effect on privacy of introducing DRM to this straightforward model. Before going on, it is worth asking why DRM is needed. The standard answers are that owners and distributors need some post-download control over their content (*i.e.*, that the user in the basic DRM-free transaction described above would copy, distribute, or modify the content in ways that violate copyright), that users need a way to verify provenance and authenticity of content, and that, in general, business models that are used for physically embodied works such as books and CDs will break if the works are embodied in digital files and sold through the type of straightforward purchase described above. This hypothesis may be correct, but it has not been adequately tested. Mass-market distributors have not yet put much high-quality content online and tested the proposition that, if they offer it at a reasonable price through an easy-to-use channel, their users will buy it rather than steal it. Some niche-market distributors are doing so, and the results of these experiments will be interesting. In the meantime, because DRM systems are under development and will apparently be used by big distributors, their potential effects on user privacy are important.

There are several different DRM strategies, and their potential effects on privacy differ as well. We consider them in turn. Note that we are not claiming that the DRM technology needed to implement these strategies would "work," in the sense that it could not be circumvented; we're asking what its effect on privacy would be if it did work.

One strategy is to distribute persistent, complete DRM metadata with digital content. In the basic transaction described above, each digital work available on the distributor's website would be formatted for use only by approved application programs, and each of these applications would be able to interpret the distributor's DRM metadata as well as the content. Each file downloaded by a user would include both the content and the metadata that describes the "rights" that the user has acquired. Content and rights need not be downloaded from the same website; the point is that, in this general DRM strategy, they are each transferred once to the user's computer (after payment if this is a commercial transaction) and thereafter unmonitored by the distributor. Using an approved application, the user could only access the content as specified by the rights metadata. Approved actions for text, for example, might include viewing and printing but not editing or redistributing via email. Promotional music or video might be distributed free but restricted via DRM metadata to a fixed, small number of plays. Under this strategy, rights metadata is added to the collectible, minable information about a user. Otherwise, the strategy has no effect on user privacy; in particular, *actual* use of the digital content, as opposed to approved but not necessarily executed use, can take place in private, offline.

Another strategy is to tie downloaded content to a particular device or set of devices. Before downloading a digital work, the user would have to provide serial numbers (or other IDs) of the devices on which he intends to use it. The DRM metadata distributed with the work would then have to include this set of devices, and the set would thus become yet another type of user-specific information that can be collected and mined. In addition, this strategy requires ongoing, periodic contact between user and distributor. Users who have purchased content will insist on being able to continue to use it when they replace old devices with new ones, *e.g.*, because the old ones malfunctioned or because the users bought more powerful devices or received them as gifts. Thus, the rights server, which may or may not be the distributor, will have to save and periodically update a record of the user and the device set. Users can thus be "tracked" to some extent under this DRM strategy. This is a departure from the previous scenarios in which a user downloaded some content and possibly some metadata, paid for it if necessary, and from then on did not need to have any further contact with the content distributor or rights server. Such tracking for the purpose of tying content to a set of devices obviously puts at risk some previously private information about the user.

A third DRM strategy involves tying the downloaded content to the user. After downloading a digital work from a legitimate distributor (and paying for it if necessary), the user would have the right to use the work on any device at any time, *after* proving that he's a legitimate user. Here, the amount of user-tracking can be much greater than it is in other strategies. Potentially collectible and minable information about a user includes his complete listening, reading, and viewing history. This is a qualitatively more serious threat than those previously described. Many people would be willing to risk others' knowing that they downloaded a pornographic video; fewer would want others to know that they watched this video 1,000 times.[2]

Finally, a radical and interesting strategy consists of dispensing altogether with content downloads. Some conventional wisdom has it that sale of digital content can never be profitable (in part because users will expect to pay much less for it than they have traditionally paid for physically embodied content, while distributors' costs will not be much lower than they are in the physical world) but that subscriptions to content services may be. In the "content becomes a service" scenario, paid-up subscribers to Internet content-delivery services can get any of the content in their subscription package streamed to their devices whenever they want it, but they don't actually acquire copies of the content. Such subscriptions might be attractive to users if they are reasonably priced, but they create the potential for massive collection of usage data and the resulting loss of privacy.

These are not the only possible DRM strategies, but they illustrate the basic, important point that some potential threats to user privacy derive from monitoring of content and rights acquisition, some from the need to update rights, and some from the collection of usage data.

As we explained earlier, some of the potential threats to privacy are caused by Web-based distribution and not by DRM *per se*. It is worth noting that there are also ways in which Web-based distribution is potentially *more* conducive to user privacy than older distribution channels. For example, because delivery is done via a data network, there is no intrinsic need to supply identity-exposing real-life information, such as a delivery or billing address, in order to complete a transaction. Furthermore, certain digital distribution channels, such as decentralized peer-to-peer systems, naturally make user-tracking harder than it is in other channels and could thus lead naturally to more privacy. When DRM systems are introduced in order to have fine-grained control over the usage of content, they can also dispense fine-grained anonymous payment tokens, thus allowing *usage* tracking (*e.g.*, for the purposes of efficiently provisioning networks or accurately compensating artists) without *user* tracking. More generally, DRM systems can segregate personally identifying information (PII) such as names, addresses, and phone and credit-card numbers into precisely the databases and operations systems of the distributor that actually need to make use of it, keeping it out of digital credentials, tokens, and other objects that are used by content-rendering applications and other system components that do not need PII.

3 Why cryptography is insufficient

Twenty-five years of cryptographic research has yielded a vast array of privacy-enabling technologies that support many types of two-party and multi-party interactions. Thus, cryptographic researchers might wish to believe that user privacy in content distribution and DRM is a solved problem. You pay for content with anonymous electronic cash. You connect to content providers and rights lockers via an anonymizing mixnet. You authenticate yourself with anonymous credential schemes or zero-knowledge identification protocols. You download content via private information retrieval or oblivious transfer. You use secure function evaluation when interacting with services that require some information.

Despite the apparent profusion of such technologies, few are in widespread use. Furthermore, even if they were in widespread use, they would not necessarily eliminate the user-privacy problems that we have described. In this section, we seek to understand why this is so. We first examine this issue from a technical perspective, then consider relevant economic and business aspects.

3.1 Technological failures of privacy solutions

Privacy-enhancing technologies developed in the research community are not stand-alone, adequate solutions to privacy threats. We claim that these technologies are insufficient: They solve only part of the problem and are open to technical attacks, they are often difficult to integrate with the rest of the mass-market content-distribution infrastructure, and they sometimes have unacceptably high costs.

[2] In a more realistic example, a friend of ours who is not a technology professional said "All of the Barry Manilow fans out there would be blackmail victims."

Our abstractions don't model our reality The cryptographic research community models interactions and protocols in terms of very distinct entities and information. For instance, the traditional communication confidentiality model is that Alice wants to communicate with her friend Bob without adversaries Eve and Lucifer being able to read her message. We may abstract general privacy-enhancing protocols by saying that users try to hide information by performing computations in some trusted private environment (the trusted computing base, or *TCB*) and then using the results of these computations to communicate with the outside world. We suggest that this model is inadequate for commercial content distribution, where the clean dichotomies of good guy vs. bad guy, trusted vs. untrusted, and private vs. public do not exist.

DRM comes into the picture because users want to obtain mass-market content online and commercial distributors want to sell it to them. Many users are unconcerned about the commercial distributors knowing the details of the purchases in which they participate directly and using this knowledge for straightforward business purposes (such as order fulfillment and billing), but many are concerned about how such knowledge could be misused or shared. This problem is further complicated by the fact that "misuse" is ill-defined; pairs of parties that have some interests in common also have some conflicting interests. Alliances, partnerships, and commercial relationships are constantly in flux and will remain so. Not only are users battling from the low ground, but it is difficult for them even to identity the enemy from whom they should hide all of their private data. In a network where businesses bundle valuable content and personalization services, and users want anytime anywhere access from any number of devices, who is Alice, who is Bob, who is the enemy, and what is the TCB? Cryptographic research cannot answer these questions. Cryptographic protocol specifications assume that one knows exactly what is "legitimate use" of data and what is "misuse," and they assume that there is a single, well-defined relationship between Alice and Bob.

Technical limitations: security breaches and usability Even if cryptographic research could answer these questions, attackers are likely to find technical exploits and information leakage in a purely technical solution to user-privacy problems. The research community is relatively good at designing secure protocols, but secure implementation is much harder, much more complex, and not something over which the research community has complete control. Brady *et al.* argue in [3] that very moderate attackers will always find vulnerabilities in large and complex systems: Statistics favor the attacker as such software systems evolve.

One critical technical concern with privacy-enhancing technology is the apparent tradeoff between ease-of-use and security. Even "simple" technology like encrypted email, for which toolkits such as PGP were offered in the early 1990s, is still confusing and difficult for the average consumer to use [32]. In DRM systems, managing separate pseudonyms for accessing content or interacting with separate entities for content versus rights management may prove to be similarly confusing.

Many ease-of-use issues relate to the problem of consumer authentication. Most client authentication techniques, *e.g.*, passwords and Web cookies, are relatively weak and prone to attacks [22, 14]. Furthermore, recovery techniques for forgotten passwords are arguably weaker: They are often based on something a user has (such as an physical or email address) or something he knows (such as answers to Web-form questions). These techniques generally require a large amount of PII and offer marginal security. In fact, privacy may turn out to be infeasible in some information-theoretic sense if we also want to provide recovery.[3]

These security weaknesses have privacy implications. Even ignoring business misuse (accidental or otherwise), if PII is collected and an attacker compromises a server, he may learn embarrassing or destructive information. For instance, cracking a DRM rights locker and exposing that a CEO recently purchased "How to survive after bankruptcy" will certainly not help the company's stock price. The distinction between "system security" and "user privacy" may be increasingly hard to make.

Legacy system integration Privacy considerations should be part of system design from the beginning, as should security considerations. Privacy decisions should be made when one maps out data flows and the format of data exchanges; they are hard to change after standardization or wide acceptance. Unfortunately, if businesses choose to engineer privacy at all, they will probably have to integrate privacy solutions into legacy systems, because of switching costs. System integration poses several difficulties:

– **Dual operation:** Legacy software may have to support dual modes for backwards compatibility, one that supplies information, one that doesn't. There have been classic attacks (*e.g.*, SSL version rollback

[3] That is, the user can have no private input to the recovery procedure.

[30]) against such designs. What good are advanced privacy options if a vendor does not update his software, and a user's DRM client happily supplies full information? One natural response to dual operation problems is forced updates for client software, which has its own drawbacks with respect to security, privacy, and consumer acceptance.

- **Information:** Legacy systems may expect more information than new privacy-friendly exchanges provide. It has been the experience of the authors that businesses often do not fully recognize the purpose or effect of such information. For instance, businesses may decide to replace social security numbers (SSNs) with random numeric ids in a database, without realizing until later that SSNs were used for risk management to reduce collection costs. As usual, system documentation is often incomplete.
- **Performance:** Legacy systems may expect grossly different performance from that offered by new privacy-enabling protocols. Fiddling with timers may not be sufficient for complex, adaptive protocols, especially for multi-party situations, multiple network layers, and many-node systems.
- **Load:** The increased computational and communication load of cryptographic protocols may cause a congestive collapse of networking and routing protocols, as well as client/server operations.

Excessive technical costs? There are two sets of costs in respecting customer privacy. They are the cost of deploying and operating the privacy technology and the opportunity cost of the activities rendered impossible by respect for privacy. The latter business incentives are described in Section 3.2.

The direct costs of privacy solutions include development, operation, policy-management, and performance problems caused by increased communication and computationally more expensive protocols. We show how the costs of several cryptographic technologies would apply to privacy solutions for DRM.

- **Public-key crypto is slow:** Most privacy-enabling protocols, *e.g.*, blinding or oblivious transfer, heavily use modular multiplications and exponentiations for public-key operations. This creates a significant computational load in comparison to basic symmetric primitives. Aging DES can process 13 MB per second, and the AES standard Rijndael can process 30 MB per second. With 1024-bit RSA, we can perform only 98 decryption operations per second [8].[4] We recognize that cryptographic accelerators are becoming inexpensive, with a $700 card from Cryptographic Appliances able to perform 1000 RSA operations per second. Still, users should certainly not be expected to purchase additional hardware, and large server farms may still find this cumulatively pricy.
- **SSL is slow:** Consider the cost of public-key operations in SSL. Networkshop found that a typical Pentium server (running Linux and Apache) can handle about 322 HTTP connections per second at full capacity but only 24 SSL connections per second. A Sun 450 (running Solaris and Apache) fell from 500 to 3 connections per second [33].
  Crypto often plays a large role in DRM system for content protection but mostly through symmetric-key operations. If privacy technologies require extensive use of public key primitives, *i.e.*, do not use them only for initialization, they might not only throttle the rate of new connections but also greatly reduce the maximum number of simultaneous connections.
- **Small devices are slow:** Although one trend of computers is to faster and cheaper, they also trend towards smaller and more mobile. Even if cryptography is inexpensive (computationally and otherwise) on the server side, it may add an unacceptable burden on devices in which grams of mass and minutes of battery life are important engineering goals.
- **Public-key infrastructures (PKIs) are a false panacea:** The Gartner Group reported that average in-house PKI development and installation costs around $1 million, and 80% of PKIs today are in test pilot stages and may never get off the ground [18]. These numbers refer mainly to business-to-business applications. There are very few serious PKI pilots that are trying to get certificates into the hands of consumers.
  While DRM systems will likely require a PKI for vendors and distributors, users themselves do not seem to need public keys. Privacy-enhancing technologies such as attribute certificates or private credentials [4] change all this. One technical reason that SET failed was that it required a PKI for all customers, not merely for vendors.

[4] As reported in the Crypto++ 4.0 benchmarks, run on Win2000 on a Celeron 850 MHz chip.

– **Traffic analysis may vitiate anonymous cash and credentials:** One might wish to replace credit-card payments with anonymous tokens that serve as proof of purchase. This may not work very well with practical ecash schemes, because consumers will often buy a token and then immediately redeem it for the rights or the content to which it entitles them. The computational expense of using blinding or some other cryptographic technique to create the token will buy us nothing, because observers can link these two actions by the consumer and learn exactly what they would have learned from a credit-card transaction. Forcing the consumer to delay redemption of the token would probably be a poor business decision. Convenience and ease-of-use trump security for most consumers; there is little reason to expect that the same won't be the case for privacy. The traffic-analysis problem would be mitigated if ecash were widely used and accepted.
– **Mixnets absorb bandwidth:** Ignoring vulnerabilities to complex traffic-analysis attacks or simple JavaScript and cookie exploits, mixnets are ill-suited for mass-market content distribution for another reason: At best, bandwidth use scales linearly with the number of hops. Even worse, widely deployed mixnets may greatly randomize traffic patterns, as messages are thrown across North America, Europe, Asia, and Africa in order to cross jurisdictional lines. This randomization works counter to network-load balancing and current infrastructure deployment (*e.g.*, the pipes to Africa are much smaller). Protocol redesign is prohibitively expensive.

These are merely a few examples of the technical costs of privacy-enhancing technologies. In describing them, we do not mean to suggest that these technologies can never be made efficient and cost-effective; on the contrary, we encourage R&D efforts to make them so (and we participate in such efforts ourselves!). Rather, our goal is to point out the harsh reality of current gaps between real-world efficiency requirements and real-world performance of techniques that satisfy the definitions of "efficiency" given in the research literature.

3.2 Economic aspects of privacy engineering

The major constituencies involved in a privacy-enabling protocol or system must be willing to sacrifice the information that may normally be collected about the other parties or their inputs. However, in e-commerce transactions, these constituencies have conflicting interests and asymmetric power. Why should a powerful content provider wanting to learn information about his users agree to run a protocol that deprives him of this very information? The industry is likely to the follow the "Know your customer" mantra.

Many of the problems facing privacy-technology adoption can be framed in microeconomic terms: network externalities, asymmetric information, moral hazard, and adverse selection. Shapiro and Varian expose the role of incentives in the information economy in [26]. Ross Anderson poses a similar argument in [1], relating the lack of information *security* to perverse economic incentives.

Network externalities The utility of privacy technologies on the Internet may be related to *Metcalfe's law*, which states that the usefulness of a network is proportional to the square of the number of nodes. The result is that networks can grow very slowly at first but then rapidly expand once a certain size is reached. This growth pattern is not limited to communication systems, such as the telephone network or the Internet, but is applicable in many situations in which a number of parties need to coordinate investments for a new system to take off. Television, credit cards, and recently DVDs have faced such startup difficulties.

It is easy to see that many privacy technologies obey Metcalfe's law and therefore exhibit network externalities – their marginal value to a user increases with their expected number of users. Anonymous file-sharing systems will become truly beneficial to users only when a large array of content can be readily, easily accessed. Anonymous email is unidirectional (and therefore less useful) unless both parties use the anonymizing network [21]. The anonymity offered by such a network is bounded by the number of users. Similarly, electronic cash will only become useful if many merchants will accept it. We may infer from this that DRM systems are unlikely to push the acceptance of cryptographic ecash but rather will continue with existing technologies, *e.g.*, credit cards. As we design DRM systems for practical use in the near term, we should therefore expect that vendors will learn who is paying how much.

Several other features of network economics are of particular importance. Technology often has high fixed cost and low marginal costs, and switching costs for infrastructural technologies are also quite large, leading to lock-in. Assuming that corporate entities using DRM systems make decisions motivated primarily by profit (and that a good reputation for respecting customers' privacy has a measurable positive impact on profitability), these entities should only switch infrastructural technologies if the expected net present value of the benefits of switching is greater than its costs. Experience shows that this makes infrastructural switching rare, slow, and painful. Consider, for example, the nonexistent migration from IPv4 to IPv6.

Often, part of what makes a business an "Internet business" is that it can use pre-existing Internet infrastructure to get a cost advantage over its competitors. If privacy technologies require widespread infrastructure redesign, they vitiate this principle of Internet business success, and content providers probably won't adopt them. If ubiquitous onion routing requires changing network protocols and routers, and the only benefit is consumer privacy, we had better not have to wait for onion routing to be in place in order to be able to buy and read e-books in private!

Once again, we are not suggesting that Internet infrastructure will never evolve to a state in which it seamlessly and efficiently incorporates privacy-enabling protocols. Our point is that such a state is quite far from the state we're in now, that evolution will be slow, and that the desire for user privacy in DRM may not be sufficient motivation to force infrastructural change. Interim steps are needed to ensure reasonable privacy options in today's infrastructure.

Asymmetries, moral hazard, and demand An asymmetry of information between entities in a DRM system makes privacy more difficult to achieve. Moral hazard arises from the principal-agent problem, in which the principal (*i.e.*, consumer) cannot observe the effort of the agent (*i.e.*, content/service provider) and thus has to incentivize the agent using something other than a payment per unit of effort. The hazard arises when the results of the agent's effort (*i.e.*, the "amount" of privacy) cannot be measured accurately, and thus the agent is tempted to slack off.

The obvious conclusion of this economic argument is that content providers will be tempted not to provide privacy if providing it costs money (and we have just argued that it does), and consumers cannot really measure their "units of privacy" and make educated demands.

Consumers are largely unable to differentiate between privacy options, and there are not even good methods for evaluating privacy. "Best practices" for privacy engineering have not yet been standardized. It is noticeable that Earthlink launched a new $50 to $60 million ad campaign, focusing strongly on offering "the totally anonymous Internet" [24]. In reality, this means they promise to abide by their privacy policy and will not share individual-subscriber information. What is the technical difference from many other ISPs? Likely none. How does the average consumer differentiate this "trust me" approach from technological ones such as the Zero-Knowledge Freedom network [15], which provides pseudonymity based on cryptographic onion-routing [27]? Likely poorly, notwithstanding greater latencies. Even if businesses decide to offer privacy, this consumer inability to differentiate motivates companies not to invest in expensive technological options.

Business incentives Two main issues face businesses that are considering privacy practices: why they *should* collect information and why they *should not* offer privacy.

There are legitimate reasons for businesses to collect data, such as customer retention, statistics, risk management, customization, and billing. For instance, network operations can (and perhaps should) collect usage data for traffic-modeling and provisioning purposes. Lack of good Internet traffic models is a big problem, and Internet-traffic modeling is a very active area of research; it requires the collection of usage data. In the content-distribution and DRM space, network operators would want to know which content is accessed from where, especially in rich media formats, in order to distributively cache replicas for bandwidth-saving, latency-reducing, and load-balancing purposes (an approach taken by Akamai for dynamic content routing [20]). In a subscription-service model, content providers would still need to know how often a song is accessed in order to determine artist compensation. For risk-management purposes in DRM systems, businesses want to be able to blacklist compromised devices or revoke compromised public keys. Similarly, most payment mechanisms require additional information to mitigate fraud, *e.g.*, asking for billing address information for online credit-card payments.

Businesses also have incentives not to offer privacy, in addition to the value of the information itself. Information security is difficult and expensive. Businesses still spend large amounts of money and many person-hours trying to achieve it, because it protects their own interests. Information privacy seems to be comparably difficult, similarly requiring secure design, implementation, and operation. However, businesses do not have the same incentives for privacy, and this results in little spending for technological innovation and development.

However, there are also motivations for minimizing the information collected. One of the strongest reasons, regulation, is concerned with both compliance and avoidance. Companies must comply with regulations such as the E.U. Data Protection Directive, and a large number of laws have been proposed in the U.S. Congress and state legislatures, as Americans grow increasingly concerned about their privacy. Businesses are starting to see that collecting and correlating data can present a public-relations risk. Lastly, extensive data collection and distribution can substantially increase the cost and effort that a business must undergo in order to be audited. We return to these issues in Section 4.3.

To summarize, it is likely that, in content distribution and DRM, as in many other realms, businesses will fight tooth and nail for the right to collect information they deem necessary. Depending on the nature of the information, consumers probably will not fight as hard to prevent them from collecting it. One major consideration for practical privacy engineering is that information *collected* for legitimate purposes may also be *used* in illegitimate ways, including sharing it with third parties who do not have a need to know. In Section 4, we recommend several steps that businesses and DRM-technology developers can take to limit the extent of this sharing and the damage it can cause.

4 Approaches to practical privacy engineering

Our recommendations fall into two general categories: (1) Fair Information Principles and ways to implement them and (2) the need for privacy audits and privacy-policy enforcement.

4.1 The Fair Information Principles approach

We have argued that definitions of "privacy" found in the cryptographic research literature are inadequate for most real-world privacy-enhancing solutions. If one accepts this claim, what should be the goals for practical privacy engineering? The best general answer we can give today is the Fair Information Principles (FIPs) [31], an early and commonly used framework for examining information-collection practices in privacy-sensitive areas such as health care. The FIPs have been widely accepted as describing desirable privacy goals. Variants of these principles underlie most privacy-protection laws and principles, *e.g.*, European privacy legislation [12, 13, 29].

The OECD version [12] of the FIPs is the following:

- Collection Limitation
- Data Accuracy
- Purpose Disclosure
- Use Limits
- Security
- Openness
- Participation
- Organizational Accountability

These are useful guidelines, in part because they do not specify any technological approach but rather set general goals. This differs significantly from the cryptographic approach, in which defining the terms of information-theoretic privacy or computational indistinguishability almost automatically suggests a technological approach, leading us to use expensive cryptographic tools such as public-key encryption, blinding, zero-knowledge protocols, *etc.* The FIPs allow us to consider low-cost solutions for privacy-enhanced electronic-commerce technologies.

This is important in light of many of the reasons that *businesses* have not widely adopted privacy-enhancing technologies (as described in Section 3.2). Even if the R&D community makes great strides on some of the problems pointed out in Section 3.1, it is unclear how quickly this technology would be developed and adopted; privacy may remain a low enough priority for the major constituencies to make serious investment unlikely.

On the *consumer* side, privacy studies continually report that consumers are very concerned about privacy. Yet today, we do not have evidence that consumers have broadly adopted software to enhance their privacy, *e.g.*, cookie blockers. Asking consumers to install and learn new, privacy-respecting software has been largely unsuccessful: The average user apparently does not want to pay for this stuff, in the broad sense of the word "pay." Rapid adoption of new technologies on the Internet is driven by the next "killer app," *e.g.*, Napster has done a lot to introduce consumers to digital music. Privacy does not fall into such an exalted category.[5]

Therefore, we suggest an alternative approach to privacy engineering that avoids some of these pitfalls:

1. The Fair Information Principles are an adequate notion of privacy.
2. Privacy enhancement should be built directly into the DRM technology that powers consumer applications. Consumers should not be burdened with needing to take additional steps to protect their privacy.
3. The business costs of introducing privacy enhancement into DRM should be low.
4. The consumer costs of using privacy-enhanced DRM should also be low. These costs include both the monetary cost of the service and the ease-of-use, latency, and other "user-experience" issues.

Why the FIPs apply to DRM One may view DRM technology as security middleware. It is typically targeted towards international markets, *e.g.*, towards European and Asian, as well as American, jurisdictions. One of the most compelling arguments for the FIPs is that they already underlie most privacy-friendly legislation and best-practice standards. Businesses that operate in (or plan to expand into) these jurisdictions will minimize their compliance costs if DRM technology can be easily configured to comply with these regulations. Furthermore, because the FIPs are emerging as the *de facto* measure for good privacy practices, business PR needs may largely be satisfied by complying with them.

The Fair Information Principles clearly stress the notion of the purpose for which information is collected and used. This is particularly well suited for relatively complex systems like DRM, in which there are a number of legitimate purposes for collecting and using information, as pointed out in Section 3.2.

4.2 Simple principles for privacy engineering

Although the FIPs are well understood, the technological literature has said relatively little on how to translate them into engineering principles. In this section, we describe some system-architectural, system-engineering, low-tech, and no-tech principles that begin to allow one to meet these goals for content distribution and DRM.

Customizable Privacy Many businesses may deploy DRM middleware, with possibly different preferred information-collection and privacy policies. This makes DRM systems different from specific, unilaterally deployed e-commerce solutions. A DRM system should therefore offer *customizable privacy*, within which system participants can easily configure the system to accommodate their preferred information-collection and handling procedures.

This customization principle has several simple consequences. A system should work with *minimal data exchanges*, and personally identifying information should not be included by default. Creating additional channels for information flow is significantly easier than trying to remove existing information flows. A first step in privacy-aware system design is to analyze the need for information, to graph flows among the various system participants, to analyze how the information flow can be minimized, and to design the message formats accordingly. This point further demonstrates the following, of no surprise to security engineers:

> Privacy considerations should be part of the initial system design phase. They should not be considered a property that can be added on later.

[5] An analogous phenomenon has been observed in the security field.

Collection Limitation A business needs to determine which information is necessary for business practices and legacy-system integration, as well as the purpose for this information. For example, credit-card security requires the transfer of billing address information. However, many applications may not require PII. In DRM systems, one should give special consideration to collection-limitation issues for the activation and individualization of DRM clients, during which these clients are endowed with public/secret key pairs. (One main purpose of individualization is to ensure that compromised devices can be identified and revoked; this does not necessarily mean that individual users must be identified.) Statistical purposes, *e.g.*, user profiling, recommendation services, and customization, similarly do not require full disclosure of PII. A unique identifier or pseudonym can be used to link the appropriate information together.

> A business should only collect information that it really needs and should disclose how this information will be used.

A business can avoid a "vacuum cleaner" approach to information collection by analyzing its real importance. Certainly this analysis is difficult, especially within the realm of a new technology such as DRM. But, as *The Economist* points out in [11], "Firms are more likely to collect information sensibly and thoughtfully if they know why they want it."

Database architecture and management Database design offers a prime opportunity to provide a layer of data-privacy technology. In fact, database design will affect the set of privacy choices businesses can offer their customers. Data may be segmented according to the different groups that are interested in it – a principle of *split databases* and *separation of duty*. For example, accounting needs customer names and billing addresses, and customer service may need some proof of purchase to verify warranty validity. Marketing and risk-management departments, which may require usage data, can function with only a pseudonym. This weak *pseudonymization* is likely to be a simple pointer into another database, but this separation may aid privacy audits, simplify sharing arrangements, and provide an easy means for access control within an organization.

> A DRM system should provide easy pseudonymization that can be used to key databases.

According to good collection-limitation practices, some DRM systems may not require a user's name.[6] In those systems that do not require full PII disclosure, splitting usage data from billing data is even simpler; there may be no need to manage a secure mapping between separate databases.

The practice of *data erasure* should also be performed as a privacy-friendly data-management technique. Data fields that contain PII should be erased after their immediate need has been fulfilled. The removal of PII from usage records before those records are inserted into a long-lived data warehouse can definitely be done efficiently on a massive scale; in fact, it is *already* done efficiently on a massive scale.[7]

Purpose Disclosure (Notice) Several considerations must be taken into account for purpose disclosure: a means to express the relevant practices in an *understandable* way, a channel to transport these various choices to the user at the *proper* time, and a channel to transport the user's decision to *other* system participants that need to observe these choices. For example, banks have complied with the letter of the Gramm-Leach-Bliley Financial Services Modernization Act, but not its spirit, by sending out notices that are densely worded and incomprehensible [19].

> Notices should be easily understandable and thoroughly disseminated.

In the DRM world, a consumer should be notified about privacy practices that accompany a certain content offer before any actual purchase. In the current DRM world, content acquisition typically occurs

[6] In fact, given that experienced or professional troublemakers are likely to engage in identity theft before buying their devices or services, not requiring a name prevents anyone from laboring under the false expectation that they can track down and punish troublemakers.

[7] We have personal experience with the responsible use of depersonalized telephone calling records for traffic modeling and other information-sciences research purposes.

through a Web retailer. This provides an easy channel for notification, either by linking to a privacy policy in HTML or by automating tools for notice such as P3P [7][8]. A DRM system may want to enable several different information-collection practices, with some requiring usage-data collection and others not requiring it. For server-side data collection, a user's decision should be transferred to the rights-fulfillment server and/or content provider. For client-side data collection, we should prevent any reporting (through simple access control) to usage-clearinghouse servers unless the user has been notified – and has agreed – to such practices.

After this paper was accepted, a lawsuit was filed against Fahrenheit Entertainment and Music City Records for distributing a CD whose content was not amenable to standard "ripping" techniques. Among the major claims of the suit are that listening to the music was no longer anonymous and that this was improperly disclosed [9].

Choice One of the most difficult challenges for FIPs compliance is giving users reasonable choices for information collection. One of the reasons for this is that businesses may not *wish* to give consumers real choices; they may want to collect more information than what is actually needed to complete transactions.

Businesses should attempt to minimize bias when presenting consumer privacy choices. In public policy, there is a known and studied phenomenon whereby a planning organization presents a set of options that are all roughly similar, thus allowing the appearance of debate (respecting choice) while knowing what the outcome will be. The consequences of various choices should be made readily apparent. We note that businesses already seem to think that simplicity is an effective differentiator in the service industry: Advertisements for "$0.05 anytime, anywhere" long-distance phone service arose from consumer frustration with hidden and exploding phone rates.

Another wrong way to approach privacy is to offer no reasonable choice at all, such as offers for "(1) free subscription service with usage-data collection, (2) gold-star privacy service for $20 per month," or worse, "(1) no privacy, (2) no service." The majority of consumers will seek the cheapest type of service, leading to no real privacy enhancements, and businesses will still incur the capital costs of implementing and deploying privacy-protected service for the small minority. On the Internet, there have been few situations in which consumers have been willing to pay for privacy.

Client-side data aggregation Client-side data aggregation can provide low-tech privacy enhancement for statistics-based services, as in profiling or recommendation services. The granularity of reported usage data will automatically affect the amount of privacy that a user enjoys. If profiling can be based on categorization and "binning" techniques[9], then users can aggregate data according to simple categorization criteria: "Last week, accessed 30 blues albums, 16 classical; played 14 hours rock, 16 hours hip-hop ...".

Transferring processed data Many of the data flows in a DRM system will not need to be "complete." For example, an organization such as ASCAP does not need to know who listened to a given song in order to distribute royalties, only that an aggregator will properly pay. There may be audit requirements to ensure that all payments flow through, but those can be accomplished in a number of ways that do not require the sharing of personal information. This is similar to the client-side aggregation suggestion, but the disclosure of data from the consumer to a counterparty makes it qualitatively weaker and requires greater trust.

Competition of services Competition in the content-provider and distribution market generally motivates service providers to offer better services to consumers, the traditional argument for a free market economy. Thus privacy could become a distinguishing feature among the offers of various service providers. Unfortunately, content ownership is largely restricted to a few organizations: 80% of music is controlled by five parties, and a few huge Hollywood studios control most of U.S. (and world) video. We believe that a liberal licensing model to competing content distributors would best suit consumer needs.

[8] We consider P3P in greater depth in Section 4.4.

[9] For example, a Beatles, Gerry and the Pacemakers, or Kinks album can all be placed in one "British Rock" bin, which similarly protects our Barry Manilow-loving friends.

Keeping business interests in mind DRM demonstrates another important phenomenon for privacy engineering. There are many components in the system – Web retailers, content servers, rights-fulfillment servers, lockers, usage clearinghouses, *etc.* – that may be operated by different entities. A user interacting with various participants will disclose different types of information and PII. The *overall* privacy that such a system provides is upper-bounded by the privacy that any one entity provides: The overall privacy is as strong as the privacy offered by the "weakest" link. To make privacy-engineering efforts effective, it is essential to understand the business interests of the various system participants. Misguided privacy efforts may simply *not* improve the level of overall privacy. If a party's business model is based largely on datamining, that party will reject restrictions on its data-collection practices.

This phenomenon demonstrates that system privacy is as much a policy issue as a technological one. Certainly, a DRM technology provider can make privacy standards mandatory for parties involved, but their effectiveness will depend upon the leverage that this technology provider can exercise. It is much more likely that overall privacy features of a DRM system will be determined by the powerful interest groups, *e.g.*, the content providers and large distributors on the one side, and consumers (or, more precisely, consumer-interest groups and privacy-activist groups that have effective media leverage) on the other side.

This leads to our key argument about how DRM may become a key enabler of privacy in the content-distribution space. DRM is conceived to support "real" content business: the exchange of money for access to and usage of content. This provides payment for content creators and other parties in the value chain, *without* their having to resort to "free" business models that are typically supported by targeted advertising and datamining revenues that most likely involve privacy intrusions.

4.3 Enforcement and auditability of privacy solutions

One may argue that the FIPs approach is "weaker" than cryptographic privacy. This is certainly true in a purely theoretical sense; it is not "provably strong." Adherence to the FIPs indeed requires honest behavior by the party obtaining the information, as well as the initial goodwill decision to offer a strong privacy policy, although this is certainly coupled with a concern for reputation and legal compliance. However, we note that running cryptographic protocols also requires the goodwill of the parties involved. Cryptography only enforces practice *after* this goodwill decision to run the relevant cryptographic protocol has been made. Thus, the (only) essential parts we are missing in a relatively "cryptography free" implementation of the FIPs are effective mechanisms to enforce information-collection practices that have been agreed upon.

The combination of notice and auditability is strong. Arguably the worst thing U.S. companies can do with regards to handling customer information is to be exposed violating their advertised privacy policies. In some cases, this has triggered class-action law suits and Federal Trade Commission investigations [10]. Along these lines, we believe that the FIPs do actually provide consumers with relatively strong privacy assurances. Requiring or incentivizing companies to make their privacy policies public – already mandatory under European privacy law – is an important step towards providing users with strong privacy. Some companies are already taking such a privacy-friendly approach: All websites on which IBM advertises are required to post a privacy policy, as are all sites that use Microsoft Passport authentication services.[10] The integration of P3P into Internet Explorer 6.0 may turn out to be another driver for websites to post privacy policies.

The combination of notice and auditing would certainly be stronger if tools were available to more effectively ensure that companies actually follow their privacy claims. For this purpose, we describe some useful principles for the enforcement and auditing of privacy practices. At the low end of the auditing solutions (actual costs to businesses that are below $10,000 [2, 28]), we have trust seals such as BBBOnLine and TRUSTe for privacy policies themselves. These services rate privacy policies and ensure that they simply and understandably state which PII is gathered, how this information will be used, with whom it is shared, which choices are available to users, *etc.* At the high end, major auditing firms audit the privacy practices of corporations. These auditing costs have reportedly been in the $2M to $5M range for large corporations [23].

We note that virtually all of these large corporations fail these rigorous audits [17]. Many of the problems result from "process" issues. The collected information is used by many separate parts of the company;

[10] We do not try to address here the privacy problems posed by a service like Microsoft Passport itself.

tracking where and how information moves around is difficult. Enterprise privacy auditing would be facilitated by keeping comprehensive access logs for databases containing PII. Logging itself must be carefully designed, however, to not reveal PII, and it should be similarly secure. These failures support our points that PII flow should be simple and minimized. Cryptographic approaches alone are unlikely to solve these process issues.

In fact, facilitating and standardizing qualified audit methods could lead to cost savings for businesses. This is another example of how businesses currently lack (and need) a good understanding of how PII should be handled, both internally and externally.

We expect that privacy-auditing technologies will fit into any satisfying and practical solution for user-privacy concerns. The purpose for which data is used is at least as important as whether it is collected. Auditing practices have proven successful and adequate in other sensitive business areas, such as financial controls. And selling privacy audits and supporting technologies may be much easier than selling cryptographic protocols, because upper management in large corporations is much more familiar with audits than with cryptography.

4.4 Adding higher tech solutions

Although we devoted Section 3.1 to pointing out the limitations of cryptographic technologies, we revisit some simple higher-tech approaches that may make FIPs compliance easier.

Proxies Straightforward third-party data proxying can be used to regulate and enforce data-collection practices. We can already note such "trust me" solutions, with Anonymizer.com and SafeWeb for anonymized Web surfing. The important property is a clear separation of duty and enforcement procedure. The proxy can verify that the user received notice and agreed to provide such information before passing the data (or some subset of it permitted by the privacy-policy agreement) on to the requester. Similarly, we can help ensure that relevant data are disclosed only to an appropriate party through the use of proxied serial connections, *e.g.*, via HTTP posts and redirects. This collection-limitation approach (through a trusted third party that provides some seal of approval) may be preferable and more justifiable than audits that happen *after* data collection.

P3P While notice and choice principles sound simple and straightforward, they are relatively difficult to engineer in an easy-to-use, transparent fashion. In a general sense, we wish to create an abstract framework that allows one to map the plethora of privacy practices into some standard framework that can be handled in an automated, integrated fashion. The Platform for Privacy Preferences (P3P) is attempting to achieve this very goal. We should consider such an approach from two different viewpoints: system capability and language expressibility.

A tool's capabilities affect the complexity and type of system that one can handle with it. The P3P specification indeed appears to be sufficiently capable to handle privacy notice within even complex DRM systems: Privacy-policy references can be dynamically generated, including references to external partner sites. Consider the desired property that users be given notice *at a proper time* of all policies that may apply for the lifecycle of a transaction. One (advanced) implementation of the P3P spec is the following: A user downloads the list of relevant policies (included in the Web retailer's HTML page), his browser automatically contacts the relevant rights-fulfillment and content servers for their policies, then his P3P-enabled browser performs some logical inferences to determine whether the policies comply with his expressed privacy preferences. One large downside of this P3P model is that the complexity devolves onto the client. One alternative is for the Web retailer to prefetch all the relevant privacy policies, compute their privacy impact itself, and present only one unified policy to the user. While this level of indirection would simplify P3P client requirements, it would add complexity and policy-synchronization requirements to the system backend.

Language expressibility impacts how deeply or granularly specific privacy practices can be represented in some standard form. Relevant to this paper: Can P3P adequately express DRM data collection and usage? Indeed, many of the possible uses of DRM data (as listed in Appendix A.2) map well into P3P purpose classifications. One major difference, however, is that P3P is precisely built for Web-based activity, not hybrid systems such as DRM. Usage data and other information generated on the client-side, as opposed to on-site, do not have any obvious coverage under the P3P statement and purpose disclosures. Furthermore, we can only classify third-party recipients of data into rough groups according to the equivalence of their data practices. We cannot easily identify recipients in a machine-readable format, even though we may wish to express very specific relationships between business entities in a DRM system. In short, P3P's expressive capabilities are somewhat inadequate for our needs.

Microsoft's adoption of the platform in IE 6 is a step in the right direction, as consumers do not have to take substantial steps to protect their own privacy, especially given default settings that block third-party cookies that use PII without explicit consent. However, Microsoft's implementation only considers compact policies, which in P3Pv1 only contain policy information related to cookies. Not supporting full privacy policies with rich user preferences, this deployment might mislead users into believing that their privacy is already being adequately protected.

P3P seeks to automate and integrate privacy-preference notice into normal Web activity. While P3P may be sufficient for DRM from a systems-engineering perspective, various sources have challenged its ability to provide real privacy protections. For instance, EPIC and Junkbusters have criticized P3P for not complying with fair information practices, for providing only a "take it or leave it" flavor of choice, and for not establishing privacy standards [6]. We remain hesitant to believe that P3P in its current incarnation will lead to actual improvements in privacy, either for DRM systems or for Web privacy in general. We fully admit general-purpose standardization is a hard problem; perhaps one may conclude that DRM languages should themselves include privacy expressibility.

Monitoring tools Note that the Fair Information Principles put a high emphasis on monitoring and restricting how collected information is actually used. Recently, companies such as IBM, Zero-Knowledge, and Watchfire have begun to build tools that help automate portions of the FIPs. For example, IBM's Tivoli SecureWay Privacy Manager is designed to monitor and enforce security and privacy policies, Zero-Knowledge's PRM Console discovers and analyzes databases containing personal information, and Watchfire analyzes websites, where much data collection takes place. We believe such tools will play an important role in privacy-policy enforcement and auditing in the future.

5 Outlook

The technical community widely accepts that building secure systems is difficult. However, there has been an enormous amount of practical and theoretical work done in the last 25 years to improve this situation. An organization that wishes to build a secure system has a large pool of resources and a number of options at its command: educating its engineers through books and professional courses, hiring experienced consultants and contractors, buying and integrating existing security products, or using standard security solutions.

Privacy engineering does not enjoy this wealth of available resources. There is great interest in privacy, but most work to date has focused on theoretical research; most actual solutions exist only in a lab context and have not been tested in large-scale deployments. A system designer building a privacy-enhanced system is very much on his own. There is no good and practical book to use; most privacy consultants focus on policy issues rather than technology; and standard software solutions or tools for development and system integration are not available.

We believe that the R&D community could make its largest contribution through the development of a *practical methodology for privacy engineering*, involving procedures for the analysis of privacy-relevant aspects of a system. This paper shows that developing such a methodology even for the subproblem of DRM systems is quite challenging. This methodology should involve a list of dos and don'ts for privacy engineering, guiding principles for white-board design, standard suggestions about how naming and pseudonymization can be handled, and the tradeoffs among various design decisions.

However, such a methodology may be a long way away. There are complex technical and social questions that are implied by the phrase "*practical methodology for privacy engineering.*" The social issues revolve around the need to define privacy so that the engineering issue can be judged. However, there are multiple definitions, many of which are mutually incompatible (*i.e.*, information self determination vs fair information practices). Privacy is a highly emotional and important topic for many people, and what information is considered private may differ substantially from person to person. Designing a practical engineering methodology that addresses all of these issues is challenging indeed. Because privacy means different things to different people in different situations, designing a *single* technical set of recommendations for handling all of them may be an unachievable goal.

The difficulty of defining requirements aside, we claim in section 3 that not all of the issues that prevent the deployment of privacy are technical. However, many engineers and technologists are deeply concerned about privacy issues. As such, the first and perhaps most achievable value of such a methodology could be in helping those concerned address the real issues preventing us from building systems with privacy.

This paper takes a step in this direction, in the context of DRM technology and system engineering.

References

1. Ross Anderson. Why information security is hard - an economic perspective, January 2001. http://www.cl.cam.ac.uk/~rja14/.
2. BBBOnLine. Privacy seal. http://www.bbbonline.com/privacy/.
3. R.M Brady, R.J. Anderson, and R.C. Ball. Murphy's law, the fitness of evolving species, and the limits of software reliability. Technical Report 476, Cambridge University Computer Labority, 1999.
4. Stefan Brands. *Rethinking Public Key Infrastructures and Digital Certificates; Building in Privacy*. The MIT Press, Cambridge, MA, August 2000.
5. Jason Catlett, Marc Rotenberg, David Banisar, Ed Mierzwinski, Jeff Chester, and Beth Givens. Open letter to Kevin Ryan, June 2001. http://www.junkbusters.com/doubleclick.html.
6. Electronic Privacy Information Center and Junkbusters. Pretty poor privacy: An assessment of p3p and internet privacy, June 2000. http://www.epic.org/reports/prettypoorprivacy.html.
7. Lorrie Cranor, Marc Langheinrich, Massimo Marchiori, Martin Presler-Marshall, and Joseph Reagle. The Platform for Privacy Preferences 1.0 (P3P1.0) Specification, W3C Candidate Recommendation, December 2000. http://www.w3.org/TR/P3P/.
8. Wei Dai. Crypto++ 4.0 benchmarks. http://www.eskimo.com/~weidai/benchmarks.html.
9. Complaint, DeLise vs. Fahrenheit Entertainment, No CV-014297, Sup. Ct. Cal. Marin County, September 2001. http://www.techfirm.com/mccomp.pdf.
10. John D. Dingell, Edolphus Towns, and Edward J. Markey. Letter by House Democrats asking FTC to investigate TiVo, March 2001. http://www.house.gov/commerce_democrats/press/107ltr30.htm.
11. Economist. Keeping the customer satisfied, July 2001.
12. Organisation for Economic Co-operation and Development. Guidelines on the protection of privacy and transborder flows of personal data, September 1980. http://www.oecd.org/dsti/sti/it/secur/prod/PRIV-EN.HTM.
13. FTC advisory committee on online access and security: Final report, May 2000. http://www.ftc.gov/acoas/.
14. Kevin Fu, Emil Sit, Kendra Smith, and Nick Feamster. Dos and don'ts of client authentication on the web. In *Proceedings of the 10th USENIX Security Symposium*, Washington, D.C., August 2001.
15. Ian Goldberg and Adam Shostack. Freedom network 1.0 architecture, November 1999. http://www.freedom.net/.
16. Carl Gunter, Stephen Weeks, and Andrew Wright. Models and languages for digital rights. Technical Report STAR-TR-01-04, InterTrust STAR Lab, March 2001. http://www.star-lab.com/tr/.
17. Dana Hawkins. Gospel of privacy guru: Be wary; assume the worst. USNews.com, June 2001.
18. Kelly Jackson Higgins. PKI: DIY or outsource? InternetWeek.com, November 2000.
19. Mark Hochhauser. Lost in the fine print: Readability of financial privacy notices, July 2001. http://www.privacyrights.org/ar/GLB-Reading.htm.
20. David Karger, Eric Lehman, Tom Leighton, Rina Panigrahy, Matt Levine, and Danny Lewin. Consistent hashing and random trees: Distributed caching protocols for relieving hot spots on the world wide web. In *Symposium on Theory of Computing*, 1997.
21. David Mazieres and M. Frans Kaashoek. The design, implementation and operation of an email pseudonym server. In *5th ACM Conference on Computer and Communications Security*, 1998.
22. R. Morris and K. Thompson. Password security: A case history. *Comm. of the ACM*, 22(11), November 1979.
23. Stefanie Olsen. Accounting companies tackle online privacy concerns. CNET News.com, September 2000.
24. Stefanie Olsen. Earthlink promises 'anonymous' web surfing. CNET News.com, March 2001.
25. National Research Council Panel on Intellectual Property (R. Davis chair). *The Digital Dilemma: Intellectual Property in the Information Age*. National Academy Press, Washington, D.C., 2000.
26. Carl Shapiro and Hal R. Varian. *Information Rules: A Strategic Guide to the Network Economy*. Harvard Business School Press, Boston, 1999.

27. P.F. Syverson, D.M. Goldschlag, and M.G. Reed. Anonymous connections and onion routing. In *Proceedings of the 1997 IEEE Symposium on Security and Privacy*, May 1997.
28. TRUSTe. Seal programs. http://www.truste.org/programs/.
29. The European Union. Directive 95/46/ec on the protection of individuals with regard to the processing of personal data and on the free movement of such data, July 1995.
30. David Wagner and Bruce Schneier. Analysis of the ssl 3.0 protocol. In *2nd USENIX Workshop on Electronic Commerce*, 1996.
31. Willis W. Ware. Records, computers, and the rights of citizens. Advisory Committee on Automated Personal Data Systems, July 1973.
32. Alma Whitten and J.D. Tygar. Why johnny can't encrypt. In *USENIX Security*, 1999.
33. Tim Wilson. E-biz bucks lost under ssl strain. InternetWeek.com, May 1999.

A A generic architecture for a DRM content-distribution system

In this section, we describe a generic DRM ecommerce architecture, shown in figure 1. We focus on giving an overview of properties and operations that are relevant for DRM operation and are important for our privacy considerations, but we do not optimize this generic architecture for privacy. Instead, we simply describe some of the key components of a DRM system and how they interact, in order to provide readers with some intuition for the terms and notions used throughout the paper.

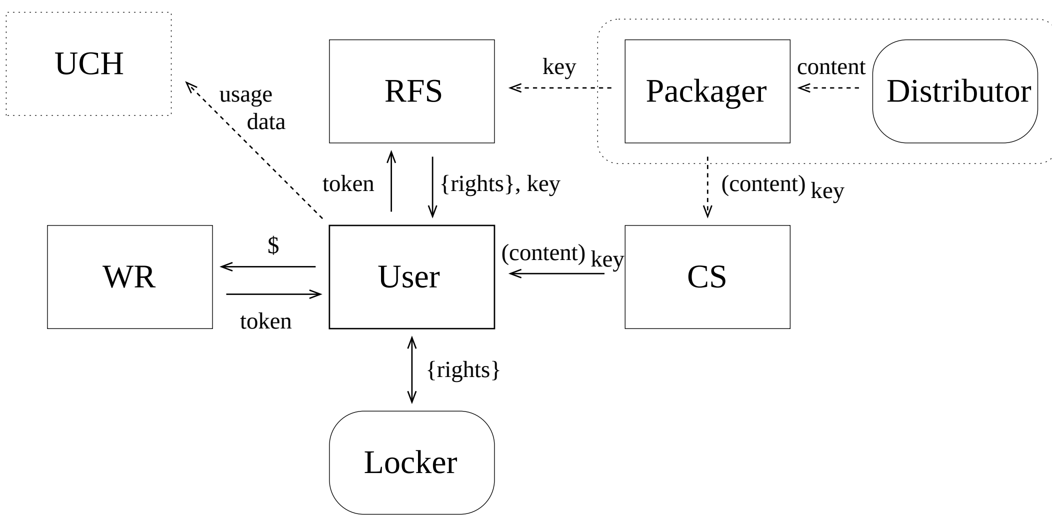

Fig. 1. Generic DRM architecture for content distribution

A.1 Basic architecture and extensions

Before content is actually distributed to a user, content is prepared by a *Packager*. This process involves encoding the content in a certain media format and encrypting it under a (symmetric) *content key*. The Packager also adds a content header to the file that contains metadata that specify the content (*e.g.*, specify the name and the artist of a song) and also hold additional information such as a URL from which the content key can be retrieved (or where the rights to play the content can be obtained). The Packager sends the content key to the *Rights-Fulfillment Server (RFS)* and the packaged content to a content distribution server.

A fundamental concept underlying DRM systems is the separation of the (encrypted) content file from the *rights*. Rights express what a user is allowed to do with a content file and may be specified in a Rights Management Language (for examples, see [16]). Business models that are currently commercially relevant and that a DRM system should therefore support include the download and purchase of individual content, subscription models (*e.g.*, to a whole music catalog), pay per play (or pay per view), and a limited number of plays for preview purposes. A simple rights language could specify timeouts, the number of times that a song is allowed to be played, *etc.*

A user needs to install DRM client software on his machine. This client decrypts content and manages access to and usage of the content as specified in the rights it has received.

More precisely, this process could work as follows. A consumer retrieves packaged content from a *Content Server (CS)*. To access the content, a user will need to retrieve the rights and the content key. In a Web-based system, a user purchases a certain offer and receives a token from a *Web retailer (WR)*, *e.g.*, Amazon.com. At the RFS, a user can redeem this token (that proves his purchase) against "digital rights" and the cryptographic content key. After having received the (encrypted) content, the rights, and the keys, the DRM client will unlock and render the content as specified in the rights.

Optionally, there may be a *Usage Clearing House (UCH)*. The UCH may collect usage data from certain clients; this may include the time and the frequency with which content was accessed or other data that relate to actual content consumption. Many of our privacy considerations focus on such usage-data collection.

There are various extensions of this basic DRM model that enhance its functionality and value to users:

- **Portable Devices:** Users may have various portable devices to which they wish to transfer the content from a PC. These may include portable music players, wireless phones, or PDAs. The number of portable devices to which content can be transferred may or may not be limited.
- **Rights Locker:** A rights locker is a storage system that contains the digital rights purchased by a user. The rights locker could simply be local to devices, but could also be implemented in some centralized way for anytime, anywhere access. Many devices may wish to interact with these central rights repositories, such as wireless phones, PDAs, *etc.*
- **Peer-to-Peer systems:** The basic architecture described above is client-server. Clearly content could be searched for and retrieved also in a P2P fashion, *i.e.*, the CS would be replaced by a P2P content-distribution service. Additionally, one may also wish to distribute rights and content keys (or implement a distributed rights locker) in a P2P fashion. Distributed rights and key management has inherent security risks from a content-protection viewpoint in a P2P architecture, likely requiring high tamper resistance on the end-points and other difficult security-engineering requirements. We do not study this architecture further in this paper.

A.2 Basic protocols and operations to support

Installation and initialization of the DRM client A user needs to install a DRM client on a device, such as a PC. The activation process may involve an individualization step for generating a unique public/secret key pair for this device, which may also involve taking hardware fingerprints to tie the client to a particular device. A user may also open an account with a rights locker, and he may register using his real-world identity or some digital pseudonym.

Searching for content and content delivery The operation of searching for DRM-protected media is not much different from a privacy perspective from using a typical search engine to find other content. When a user downloads content from a Content Server, the CS could certainly log user IP addresses. Users who may wish to protect their privacy from the CS may connect to the CS via an anonymizing network. Packaged content is typically freely superdistributable so that an anonymizing step should not violate the security of a DRM system.

Content acquisition A users needs to purchase "subscription" or "pay-per-use" tokens from some vendor, typically a Web retailer. Credit cards dominate Internet ecommerce payments, and thus the Web vendor will usually learn the real-world identity of the user. The token is transferred from the Web retailer to the RFS via the user. Note that the token is supposed to be a secure proof of purchase, and so it may specify the terms of the purchase and contain a unique serial number to prevent "double spending" of the token. However, the token does not need to involve the identity of the user.

Rights delivery The user redeems his token at the RFS to receive the corresponding rights and the cryptographic key to unlock the content. For security reasons, the rights should not be easily transferable to another user. To prevent this, the DRM client should send its public key to the RFS, under which the RFS digital signs the (rights, UID) pair and returns the signed pair to the client. This UID could be specific per user or per device, thereby targeting rights to a particular user or device. The content-decryption key is similarly delivered to the client encrypted under its public key. The RFS does not necessarily learn the user's identity during rights delivery, because a user may only pseudonymously identify himself via the public key. However, the RFS could link or log these transactions and thereby obtain pseudonymous profiling.

For simplicity, we view a rights locker as a central database that stores the digital rights of users, to which a user has to authenticate himself to access his rights. Note that authentication mechanisms to a locker should be *non-transferable*, so that a user cannot publish his user name and password (or public/secret key pair) and thereby enable thousands of people to access the subscription rights that a single user has purchased. A good deterrence mechanism is to ensure that some high-value secret is also accessible through the rights locker, *e.g.*, the authentication mechanism also allows access to a user's credit card information, name, or other information that users may be unwilling to share.

Accessing and playing content When a user attempts to access content, the DRM client determines whether the necessary keys and rights are present and, if so, renders the content. Because this process occurs on the user's local device, it does not present any privacy risks.

Risk management One main goal of a DRM system is to keep unauthorized access to content, or "privacy," under a tolerable threshold. Charged with this goal, a risk management (RM) system participant, most likely connected to the RFS, would be useful to monitor activities in a DRM system to detect misuse or anomalous activities. In particular, there should be a mechanism to revoke compromised devices, *e.g.*, by having the RFS maintain a blacklist of the public keys of compromised devices. Furthermore, the RM should have a method for renewing user software, in order to update security features on the DRM client.

The RM should be able to identify suspicious patterns in the download behavior of users, especially with massive overuse, or crawl popular websites for compromised keys. Clearly, the former approach of log analysis has some privacy implications. However, we note that revocation does not require user PII, but only the public key to be revoked or blacklisted. Still, for legal remedy or simply fraud deterrence, the RM may wish to have the ability to resolve a pseudonym to a real-world identity.

Data collection in DRM systems We give some examples as to what information can be collected in a DRM system and for what purposes it may be used. Collected information may include data about content retrieval, rights retrieval, content accessing, frequency, times, access locations, *etc.* This information can be obtained by logging server-side information, or by having client-side DRM software store relevant usage data that may be later sent to a central Usage Clearing House server.

The purposes for collecting data may include the following:

- personalized use for direct marketing
- quality of service enhancement (by network operator)
- backup and archives
- aggregate (depersonalized) usage of info for marketing, *e.g.*, to discover trends in the data set, to perform datamining
- profiling (de)personalized records, *e.g.*, the RIAA (ASCAP) may wish to collect data from a subscription service, such as how often a song is listened to, in order to properly compensate artists.
- customer service and retention
- recommendation services