Revocation in The Video Content Protection System

Toine Staring and Boris Škorić
Philips Research

***Abstract*—VCPS (Video Copy Protection System) protects Broadcast Flag and copy-once content on DVD+R/+RW (Digital Versatile Disc + Recordable/ReWritable) media. It provides device revocation using a key block on the media. Based on a statistical model, we analyze the dynamics of the key block size.**

I. Introduction

The FCC Broadcast Flag [1] rules[1], the Open Cable [2] rules, as well as the DTCP (Digital Transmission Content Protection) [3] rules, are examples of compliance rules calling for encryption of consumer-made video recordings. Current video recording devices that use the DVD+R/+RW format do not have such capability. For that reason, Philips and HP have developed VCPS (Video Content Protection System) [4].

A functionality that no modern content protection system can do without is revocation. This is the ability to prohibit hacked devices (which, e.g., have their keys published) to access newer encrypted content. VCPS revocation is based on broadcast encryption [5], [6] and uses the scheme described in [7], [8]. In this scheme blank discs contain a DKB (Disc Key Block), which is stored either in the pre-groove or in the Initial Zone of the disc. The former has the advantage of a negligible disc manufacturing cost-up, whereas the latter has the advantage of a much faster read-out time. In order to compensate for the slow read-out from the pre-groove (less than 700 byte/sec at single speed) and the inaccessibility of the Initial Zone in play-only devices, recorders store a copy of the DKB in the Lead-in Zone of the disc.

It is highly desirable to get some insight in the dynamics of the DKB size over the life span of the system. For example, we would like to know if the amount of disc space reserved for the DKB is sufficient. In addition, we would like to know if the pre-groove is a viable location to store the DKB (which is the case only if the DKB is sufficiently small). The latter is especially important in case we have to deal with high numbers of software package revocations (see section III).

II. Disc Key Block

The DKB consists of a binary tree structure as shown in Fig. 1. Each node has an associated unique *Node Key.* Each leaf node has an associated *Authorization Key,* which is the *Root Key* encrypted using the leaf node's Node Key. In addition to the Authorization Keys, the DKB contains tree structure data in the form a three-bit tag field for each node. These tag fields indicate whether the node is a leaf node or not, and whether it has a left respectively right child node.

[1] At the time of this writing, we have learned that the FCC's authority to impose these rules has been challenged, causing a delay in their introduction.

A device contains a Device ID and a set of Node Keys for processing the DKB. The processing algorithm is as follows: Start at the root node. Examine the Device ID bits one by one, from left to right. If the bit is a one move to the left child node, otherwise move to the right child node. Continue until either a leaf node is reached or the Device ID bits are exhausted. In the former case, the device can decrypt the Authorization Key, and obtain the common Root Key, which is required to decrypt the video recording. In the latter case, the device is revoked.

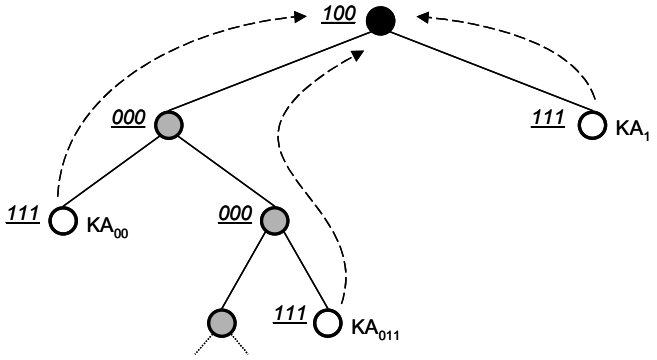

Fig. 1. Example of the top part of a DKB tree.

III. Revocation Model

To analyze the dynamics of the DKB size, we assume a stochastic model of random and independently distributed revocations. Note that this is a "worst case" situation, since it is possible to revoke continuous ranges of Device IDs very efficiently. We further assume that the number of devices sold in the market grows according to an S-shaped curve. Finally, we adopt different policies for hardware and for software "devices," which is inspired by the fact that experience tells us that software is much more vulnerable to attack than hardware. Therefore, hardware devices have a unique Device ID, and are not replaced after revocation. In contrast, all installations of a distinct software package share the same Device ID, and are immediately replaced after revocation (with a new Device ID). As a result, the number of hardware devices decreases as the number of revocations increases, whereas the number of distinct software packages is independent of the number of revocations.

IV. Results

Whereas we can calculate analytically the expected number of revocations that result from our model, this is not possible for the expected DKB size. Therefore, we have run Monte Carlo simulations to obtain the latter. From a set of time sequences, we construct histograms to determine the expected

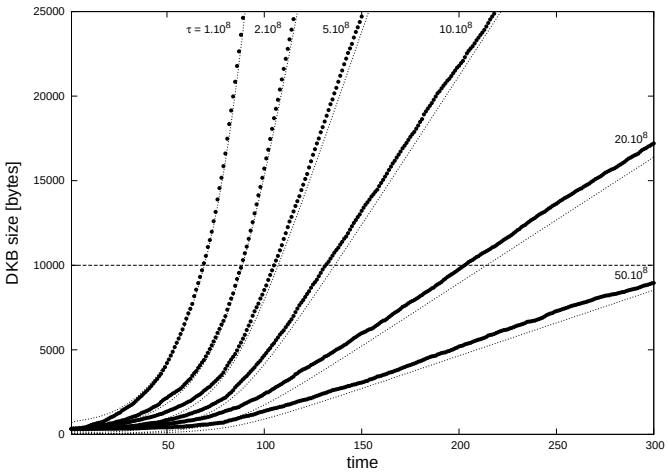

Fig. 2. Increase of the DKB size with time. Simulation (black dots) and analytic results (dotted curves) for a number of revocation time constants. The number of hardware devices reaches 10^9 after 300 time steps.

number of revocations is much higher than the number of distinct software packages for $\tau < 100$. This is not a problem in our model). We explain this result as follows. At low τ, software packages are revoked very quickly. This implies that the sets of revoked and non-revoked devices tend to segregate into two clusters, which the DKB encodes very efficiently. At intermediate τ, there is a large cluster of revoked devices as well as a cluster of devices in which about every other device is revoked. This means that the tree structure degrades to resemble a linear list, which is an upper bound to the required storage. (In a "white list" of approved software packages we would need 16 kB of storage for 1000 software packages and 128 bit keys.) Finally, at high τ, the set of revoked devices is relatively uniformly distributed over the tree, so that the DKB size asymptotically approaches the analytic result.

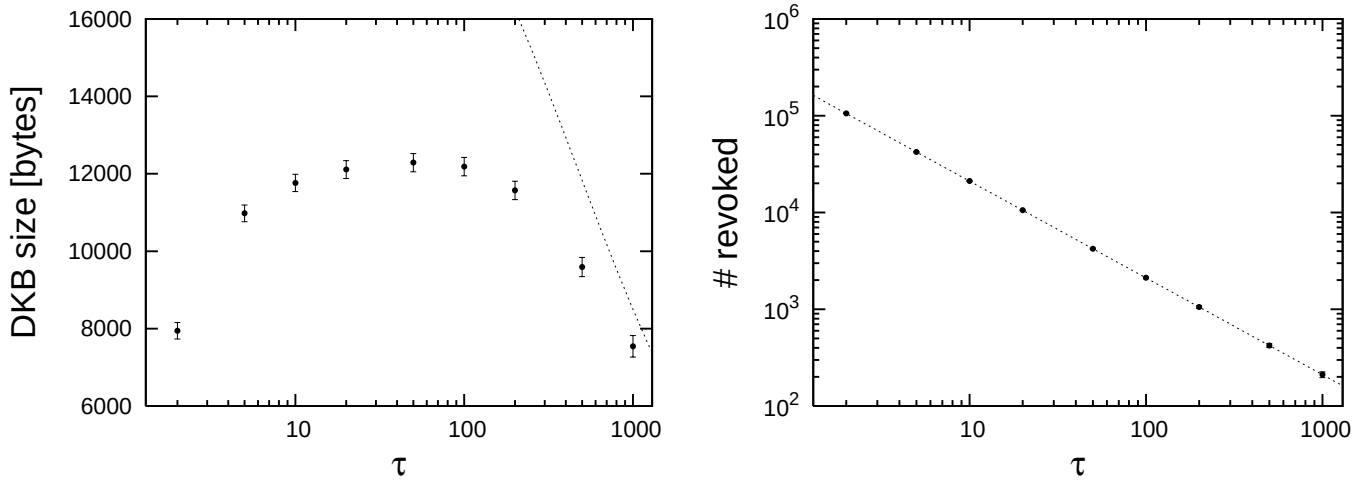

Fig. 3. Expected DKB size (left) and expected number of revoked devices (right) of software packages after 300 time steps. Simulation (black dots) and analytic results (dotted line). The number of distinct software packages is 1000.

DKB size and number of revoked devices at the end of the life span of the system. After each time step in a sequence we evaluate the number of revocations, based on a Poisson process with time constant τ. The parameters of our S-curve are chosen such that the number of devices reaches about 90% of its maximum in 120 time steps. Finally, unless indicated otherwise we reckon all time quantities in units of one month.

For revocation of hardware devices, our simulations of the DKB size follow the complete subtree result [6]: $r\log(N/r)$, where r is the number of revoked devices, and N is the total number of devices sold in the market. We thus find that the DKB can support up to about 1700 independently revoked hardware devices in the 448 kB reserved on the disc. Fig. 2 shows the expected increase of the DKB size with time. The 5% difference between the simulation and analytic results is due to tree structure and header data "overhead," which is not taken into account in the analytic calculations.

Fig. 3 shows the results of our simulations for software packages. Clearly, the $r\log(N/r)$ behavior for the DKB size does not hold in this high revocation regime. (Note that the

V. Conclusion

We find that the pre-groove is well suited to store the DKB. At single-speed read-out times below 15 seconds, it supports up to 25 revoked hardware devices or an unlimited number of revoked software packages. It should take at least a few years until pre-recording of the DKB might become necessary.

References

[1] http://hraunfoss.fcc.gov/edocs_public/attachmatch/FCC-03-273A1.pdf.
[2] http://www.opencable.com/primer.
[3] http://www.dtcp.com.
[4] http://www.licensing.philips.com/vcps
[5] A. Fiat and M. Naor, "Broadcast Encryption," *Advances in Cryptology: CRYPTO 1993,* LNCS 773, pp. 480–491, 1993.
[6] D. Naor, M. Naor, and J. Lotspiech, "Revocation and Tracing Schemes for Stateless Receivers," *Advances in Cryptology: CRYPTO 2001,* LNCS 2139, pp. 41–62, 2001.
[7] C. K. Wong, M. Gouda, and S.S. Lam, "Secure Group Communications Using Key Graphs," *Proceedings of ACM SIGCOMM '98,* September 2-4, 1998.
[8] D.M. Wallner, E.J. Harder, and R.C. Agee, "Key Management for Multicast Issues and Architectures," *Request For Comments 2627,* June 1999.