A Cryptanalysis of the High-bandwidth Digital Content Protection System

Scott Crosby
Carnegie-Mellon University

Ian Goldberg
Zero Knowledge Systems

Robert Johnson Dawn Song David Wagner
University of California at Berkeley

August 3, 2001

Abstract

We describe a practical attack on the High Bandwidth Digital Content Protection (HDCP) scheme. HDCP is a proposed identity-based cryptosystem for use over the Digital Visual Interface bus, a consumer video bus used in digital VCRs, camcorders, and personal computers. Public/private key pairs are assigned to devices by a trusted authority, which possesses a master secret. If an attacker can recover 40 public/private key pairs that span the module of public keys, then the authority's master secret can be recovered in a few seconds. With the master secret, an attacker can eavesdrop on communications between any two devices and can spoof any device, both in real time. Additionally, the attacker can produce new key pairs not on any key revocation list. Thus the attacker can completely usurp the trusted authority's power. Furthermore, the protocol is still insecure even if all devices' keys are signed by the central authority. □

1 Introduction

The High-bandwidth Digital Content Protection (HDCP) scheme is a cryptographic extension to the Digital Visual Interface (DVI) designed to prevent the copying of video data transmitted over the DVI bus. DVI is already commonly used to send digital video between camcorders, digital VCRs, and personal computers.

If the HDCP enhanced DVI standard is also adopted by monitor and television manufacturers, then it could serve as the last leg of a secure channel for the online distribution of television, movies, and other video data. Online content distributors would like to build this channel to prevent perfect digital copies by never exposing the digital video signal as plaintext in the receiver's computer.

Because DVI devices from many different manufacturers need to interoperate and perform key exchange with no user intervention, the HDCP authors chose to use an identity-based cryptosystem. It appears that the authors did not want the implementation of the scheme to be too onerous, and so avoided any conventional identity-based scheme. Instead, they designed a custom scheme that is fast, easy to implement, and insecure.

In the HDCP scheme, device manufacturers purchase HDCP licenses from a trusted authority. A license includes, for each device A, a public vector v_A, called the Key Selection Vector (KSV), and a private vector, u_A. When devices A and B wish to communicate, they exchange v_A and v_B. A computes the dot product $u_A \cdot v_B$ and B computes $u_B \cdot v_A$, and they use this as their shared secret for the rest of their interactions. The trusted authority uses some secret information to choose v_A, v_B, u_A, and u_B so that the above computations will produce the same answer. This protocol is used in both the Upstream and Downstream versions of HDCP. The Upstream version of HDCP is designed for the communication link between software running on a personal computer, such as a user friendly video playback utility, and the HDCP devices attached to that computer. The Downstream protocol is used between HDCP devices. Since the cryptographically relevant portions of these protocols are identical, our attack applies to both.

We exploit a well-known cryptographic design mistake: the shared secret generation is entirely linear. The attack only needs 40 public/private key pairs such that the public key pairs span $M \subset (\mathbb{Z}/2^{56}\mathbb{Z})^{40}$, the module generated by all public keys. Since HDCP devices divulge their public keys freely, we can easily test whether a set of 40 devices have public keys spanning M before expending the effort to extract their private keys. With these keys, the authority's secret can be recovered in only a few seconds on any desktop computer.

The consequence of these flaws is that, after recovering the private keys of 40 devices, we can attack every other interoperable HDCP device in existence: we can decrypt eavesdropped communications, spoof the identity of other devices, and even forge new device keys as though we were the trusted center. Note that this allows us to bypass any revocation list or "blacklisting": such mechanisms are rendered completely ineffective by these flaws in HDCP. Therefore we recommend that the current HDCP cryptosystem should be abandoned and replaced with standard cryptographic primitives.

The HDCP cryptosystem is also unusual in that it can be broken without fully

$A \to B:$	v_A, n_A
$B:$	$K' = v_A \cdot u_B,\ R' = h(K', n_A)$
$B \to A:$	v_B, R'
$A:$	$K = v_B \cdot u_A,\ R = h(K, n_A)$
$A:$	Verifies $R = R'$

Table 1: The HDCP Authentication Protocol

understanding its operation. The HDCP specification does not describe the key generation process used by the center but, based solely on the properties of generated keys, we can characterize all possible key generation strategies and show that they are all insecure. In other words, we can prove, given just the interface, that every possible implementation that follows this interface is insecure.

2 The HDCP Authentication Protocol

The HDCP protocol is described completely in [1]. We present an abstracted version that captures the cryptographically relevant portions of both the Upstream and Downstream versions of HDCP. A trusted authority assigns to each device, A, a public vector $v_A \in (\mathbb{Z}/2^{56}\mathbb{Z})^{40}$, called the Key Selection Vector (KSV), and a private vector, $u_A \in (\mathbb{Z}/2^{56}\mathbb{Z})^{40}$. The vector v_A consists of 20 zeros and 20 ones. The vector u_A must be kept in tamper-proof hardware or, in the case of a software implementation, obscured by code obfuscation techniques. When devices A and B wish to communicate, they exchange v_A and v_B. A computes $K = u_A \cdot v_B$ and B computes $K' = u_B \cdot v_A$. The trusted authority has used some secret information to choose v_A, v_B, u_A, and u_B so that $K = K'$.

In HDCP, one device is the transmitter and one is the receiver. To verify that the key agreement process has been successful, the transmitter A also sends a nonce n_A, and the receiver replies with the 16-bit value R' computed by $R' = h(K', n_A)$. The transmitter performs the analogous computation and verifies that the results are the same. The non-invertible function h is completely described in the specification, but the details of its operation are not important here. We assume that all DVI transmitters can interoperate with all DVI receivers, an assumption that seems to be implied by the specification.

HDCP also supports revocation of certain KSVs. Transmitters are required to check that their peer's KSV is not on the current revocation list. According to the HDCP license, KSVs can be placed on the KRL if the corresponding private key has been leaked, or if requested by the National Security Agency.

Name		Size		Comment
v_A, v_B		40 bits		Must have Hamming weight 20
u_A, u_B		Vector of 40 56-bit numbers		
n_A		64 bits		
K, K'		56 bits		$K = v_B \cdot u_{A,} K' = v_A \cdot u_B$
R, R'		16 bits		$R = h(K, n_A), R' = h(K', n_A)$

Table 2: Summary of HDCP Protocol Variables

3 Linear Algebra over $\mathbb{Z}/2^{56}\mathbb{Z}$

Since $\mathbb{Z}/2^{56}\mathbb{Z}$ is not a field, not all the basic facts from linear algebra hold in this setting. Nonetheless, much of our intuition carries over with not too many changes. In this section we set down the few results we need. Let $R = \mathbb{Z}/p^n\mathbb{Z}$, where p is prime. The following fact is used without proof.

Fact 1 *The standard determinant function,* det*, is multiplicative, and a matrix T is invertible if and only if* $\det T$ *is a unit in R. Since $R = \mathbb{Z}/p^n\mathbb{Z}$, this implies T is invertible if and only if* $\gcd(\det T, p^n) = 1$.

R has exactly one chain of ideals, $(0) = (p^n) \subset (p^{n-1}) \subset \ldots (p^1) \subset (p^0) = R$. This makes Gaussian elimination work almost as well as over a field.

Proposition 2 *Any $m \times n$ matrix A over R can be transformed, via invertible row operations, into an upper triangular matrix such that if the leading nonzero term of row i is in column j, then the leading nonzero term of row $i+1$ is in column $j+1$ or later. Furthermore, the leading terms will all be powers of p.*

Proof. The Gaussian elimination algorithm need only be modified slightly.

$$A = \begin{bmatrix} a_{1,c_1} & & & & & & \\ 0 & \ddots & & & & & \\ & & a_{k,c_k} & & & & \\ & & 0 & a_{k+1,c_{k+1}} & * & \cdots & * \\ & & & * & & \cdots & \\ \vdots & & \vdots & \vdots & & \cdots & \\ 0 & & 0 & * & & \cdots & * \end{bmatrix}$$

Let c_1 be the first non-zero column. Let r_1 be a row such that, for all r, $(a_{r,c_1}) \subseteq (a_{r_1,c_1})$. By dividing row r_1 by a unit, we can transform a_{r_1,c_1} into p^{e_1} for some e_1. We then interchange row r_1 with row 1. We can now use row 1 to cancel all the other non-zero terms below a_{1,c_1}, since the column c_1 entries of all the other rows now lie in (a_{1,c_1}). We now repeat with column c_2, the first column with a non-zero entry in rows $2,\ldots,m$, and so on. If, after swapping, entry $a_{k,c_k} = 1$, then we may optionally use row k to cancel the non-zero terms above a_{k,c_k}. It is a standard fact that the row operations used here are invertible. $\square$

Define $\sigma : (\mathbb{Z}/2^{56}\mathbb{Z})^{40} \to \mathbb{Z}/2^{56}\mathbb{Z}$ by $\sigma(v_1,\ldots,v_{40}) = \sum_{i=1}^{40} v_i$. Then, since KSVs have Hamming weight 20, for any KSV v, $\sigma(v) = 20$. Since σ is linear, σ applied to any linear combination of KSVs will be in the ideal $(4) \subset \mathbb{Z}/2^{56}\mathbb{Z}$. Since not all vectors α in $(\mathbb{Z}/2^{56}\mathbb{Z})^{40}$ have $\sigma(\alpha) \in (4)$, no set of KSVs will ever span $(\mathbb{Z}/2^{56}\mathbb{Z})^{40}$. Let M be the module spanned by all possible KSVs. The following proposition tells us when a set of KSVs spans M.

Proposition 3 *A set of KSVs $v_1,\ldots,v_{40}$ spans M if and only if the matrix V whose rows are $v_1,\ldots,v_{40}$, has* $\gcd(\det V, 2^{56}) = 4$.

Proof. Let $V' = \left[v'_{ij}\right]$ be the result of applying the above Gaussian elimination algorithm to V. Since the Gaussian elimination is invertible, there exists a matrix U, with $\gcd(\det U, 2^{56}) = 1$, such that $V' = UV$. Thus $\det U^{-1} \det V' = \det V$. Since $\det U^{-1}$ is coprime to 2^{56}, we must have $\gcd(\det V', 2^{56}) = \gcd(\det V, 2^{56}) = 4$. Since V' is upper triangular, $\det V' = \prod_{i=1}^{40} v'_{ii}$. But v'_{ii} is a power of 2 for each i, so $\det V' = \prod_{i=1}^{40} v'_{ii} = 4$. Since the only nonzero entry in row 40 is $v'_{40,40}$, we must have $v'_{40,40}$ a multiple of 4 by σ considerations. Since $\det V' = 4$, V' has the following form.

$$V' = \begin{bmatrix} 1 & 0 & 0 & * \\ 0 & \ddots & 0 & * \\ 0 & 0 & 1 & * \\ 0 & 0 & 0 & 4 \end{bmatrix}$$

Let v'_i be the ith row of this matrix. If $w = (w_1,\ldots,w_{40})$ is a KSV, then put $w' = w - \sum_{i=1}^{39} w_i v'_i = (0,\ldots,0,w'_{40})$. As we observed above, $\sigma(w') = w'_{40}$, lies in $(4) \subset \mathbb{Z}/2^{56}\mathbb{Z}$. So there exists a c such that $w'_{40} = 4c$. Hence $w = \sum_{i=1}^{39} w_i v'_i + c v'_{40}$. Note that this does not prove the existence of a KSV matrix V with $\gcd(\det V, 2^{56}) = 4$, but such matrices can easily be found experimentally. Thus the rows of the matrix V' above do lie in M.

If, on the other hand, $v_1,\ldots,v_{40}$ span M, then there exists a matrix U such that $V' = UV$. Thus, by the multiplicativity of det, $\gcd(\det V, 2^{56})$ is at most 4. By a σ

argument similar to the above, $\gcd(\det V, 2^{56})$ is at least 4. Thus $\gcd(\det V, 2^{56}) = 4$. $\square$

It will also be useful to know the probability that $40+m$ KSVs contain a set of 40 KSVs that span M. The following table was created by generating 10000 sets of $40+m$ random KSVs and testing whether the set contained a spanning subset of 40 KSVs.

Number of KSVs	40	42	44	46	48	50
Prob. of Spanning	.295	.773	.940	.982	.997	.999

4 The Authority's Secret

We now prove that the authority's secret information can be recovered by an attacker. The main insight is that the secret can be captured in a 40×40 matrix, and hence techniques from linear algebra suffice to recover it. Before we proceed, we must note that the center may choose to issue only KSVs from a submodule, N, of M, the module spanned by all KSVs.

Observation 1 *Let v be a KSV, and suppose u_1 and u_2 are both valid private keys for v. Then $u_1 - u_2 \in N^{\perp}$.*

Proof. Let (v', u') be any other valid key pair. Since $v' \cdot u_1 = v \cdot u' = v' \cdot u_2$, we have $v' \cdot (u_1 - u_2) = 0$ for all $v' \in N$. $\square$

The content of this observation is that, if two different key vectors, u_A and u'_A, form valid key pairs with the same KSV, then $K = K' = u_A \cdot v_B = u'_A \cdot v_B$ for all devices B. Hence u_A and u'_A are functionally indistinguishable.

Corollary 4 *The map $T : M \to (\mathbb{Z}/2^{56}\mathbb{Z})^{40}$, mapping public keys to private keys, is well defined $\mod N^{\perp}$.*

We can now prove that the map S has a particularly nice form.

Observation 2 *T can be represented by a 40×40 matrix, S.*

Proof. To show that a map can be represented by a matrix, we only need to show that it is linear. So let $v = cv_1 + v_2$. Then $(cT(v_1) + T(v_2)) \cdot v' = cT(v_1) \cdot v' + T(v_2) \cdot v' = cT(v') \cdot v_1 + T(v') \cdot v_2 = T(v') \cdot v = T(v) \cdot v'$, for arbitrary $v' \in N$. Thus $T(v) = cT(v_1) + T(v_2) \mod N^{\perp}$. $\square$

Recovering S is now straightforward. First collect a set of key pairs $(v_i, u_i)_{i=1}^n$ such that the v_i span N. Then use any standard technique to solve the systems of equations $U = SV$. For example, the Gaussian elimination algorithm of Section 3 can be applied here. This allows us to recover all of the trusted center's secret, no matter how it picks keys.

5 Forging Key Pairs

Let G be a matrix recovered as in Section 4. Then G and S agree on the submodule spanned by the recovered vectors $v_1, \ldots, v_n$, and quite probably disagree everywhere else. If $v_1, \ldots, v_n$ span M, then G is equivalent to S. In other words, $Gv = Sv$ for all valid KSVs v. Thus, to forge a new key pair, we can simply pick a random KSV, v, and compute the corresponding private key $u = Gv$.

The authority may try to prevent the total recovery of S by only assigning to devices key pairs with KSVs in a submodule of $N \subset M$. If $< v_1, \ldots, v_n >= N \neq M$, then we can only forge key pairs (v, u) where $v \in N$. Finding new KSVs in the span of the recovered KSVs may be difficult.[1] This could be a problem if we wish to build a device that interoperates with other HDCP devices and the authority has placed all our recovered KSVs on the key revocation list.

However, the HDCP protocol does not require devices to check that their peer's key is not the same as their own, and so a "parotting" attack is possible. To build an interoperable receiver, we can simply embed the matrix G in the device, and program it to reply to all authentication challenges with the KSV it just received from the transmitter. It can compute the corresponding private key on the fly and proceed with the authentication protocol. We can essentially perform the same trick to build interoperable transmitters, but the transmitter will have to perform two authentications. The first time, it will send a random KSV and collect the KSV of its peer. The transmitter will then abort the authentication and restart it using the KSV it just learned from the receiver.

One might be tempted to correct the defects in HDCP by signing the KSVs with a private key known only to the central authority. Then, when two devices execute the authentication protocol, they exchange the certificates containing their KSVs, verify each others' certificates using the authority's public key, and proceed as before. This change accomplishes very little. Eavesdropping would still be possible since the certificates, and hence the KSVs, of each device would be available to the eavesdropper who could then compute the corresponding private keys needed to decrypt the traffic. Devices would still be clonable by embedding the victim's

[1]It's not hard to reduce subset-sum to the problem of finding a new KSV in the span of some other KSVs. However, since the dimension is only 40, we can brute-force this problem if necessary.

certificate and private key in the clone. The parotting attack above is still available, too. The only thing certificates prevent is forging new keys. The Digital Transmission Content Protection (DTCP) standard includes a Restricted Authentication protocol that may be just such a certificate-enhanced variant of HDCP[2]. The information needed to fully evaluate the security of DTCP is not publicly available, but what little is public gives reason to be sharply concerned that DTCP's restricted authentication protocol may be susceptible to similar attacks.

6 Conclusion

These attacks are very powerful and very flexible. To recover the center's master secret, we need 40 key pairs, and we have a variety of ways to get them. We can reverse engineer 40 different HDCP video software utilities, we can break open 40 devices and extract the keys via reverse engineering, or we can simply license the keys from the trusted center. According to the HDCP License Agreement, device manufacturers can buy 10000 key pairs for $16000. Given these 40 spanning keys, the master secret can be recovered in seconds. So in essence, the trusted authority sells a large portion of its master secret to every HDCP licensee. With the master secret in hand, we can eavesdrop on all device communications, spoof any device, and clone any device, all in real time. We can produce a device that, by parroting back the KSVs of its peers, cannot be disabled by any blacklist. With a reasonable amount of computation, we can also produce new device keys not on any key revocation list. For these reasons, we recommend that HDCP be abandoned in favor of conventional cryptographic schemes.

References

[1] Intel Corporation. *High-Bandwidth Digital Content Protection System*, 1.00 edition, February 2000.

[2] Hitachi, Ltd. and Intel Corporation and Matsushita Electronic Industrial Co., Ltd. and Sony Corporation and Toshiba Corporation. *Digital Transmission Content Protection System, Volume 1*, July 2001.