ANOTHER BIRTHDAY ATTACK

Don Coppersmith
IBM Research
Yorktown Heights, NY 10598

Abstract: We show that a meet-in-the-middle attack can successfully defraud the Davies-Price message authentication scheme. Their scheme used message blocks in an iterated encipherment of an initial block, and it went through the message blocks twice, in order to prevent just such a "birthday" attack.

Background

This note concerns methods for attaching a digital signature to a long message. There are several proposals for hashing the long message into a shorter hashed value, which can then be digitally signed by a more expensive technique, for example RSA. [RSA] This allows the signature to be publicized without revealing the content of the message; it allows a shorter signature; and it decreases the computation time necessary for computing or checking signatures. [Den]

Rabin [Rab] introduced a scheme, based on a general block cipher. It can be described in terms of DES, although Rabin's proposal did not use DES. In this scheme, the message M would be broken into 56-bit blocks M_i, and these message blocks would be used as keys for the iterated encipherment of some initial value H_0. The final encipherment, along with the initial value, would form the hash value. Thus

$$\begin{aligned} &H_0 = \text{ random} \\ &H_i = E_{M_i}(H_{i-1}), \qquad 1 \le i \le n \\ &\text{RSA-Sign } (H_0, H_n). \end{aligned}$$

(Notation: here and throughout, $E_K(X)$ is the DES encipherment of the cleartext X under the key K; $D_K(Y)$ is the DES decipherment of the ciphertext Y under the key K.)

The problem with this scheme in conjunction with DES, is a "meet-in-the-middle" or "birthday" attack. The opponent, knowing the RSA-signature of the pair (H_0, H_n) arising from some legitimate message M', can devise a message M whose content is largely selected by the opponent, but whose hash value is also (H_0, H_n). Thus the RSA-signature of (H_0, H_n) can be reused to sign this bogus message.

To accomplish this, the opponent need only evaluate 2^{33} encipherments, instead of the 2^{64} required by the naive trial-and-error approach. (He also uses $2^{32} = 4 \times 10^9$ storage.) Namely, the opponent specifies values of $M_1, M_2, \dots, M_{n-2}$. Using the given value of H_0, he computes successively $H_1, H_2, \dots, H_{n-2}$. Then for each of 2^{32} trial values X for the message block M_{n-1}, he computes that value $H_{n-1}[X] = E_X(H_{n-2})$ which H_{n-1} would have if X were chosen for M_{n-1}. These 2^{32} values are sorted and stored. Now for each of 2^{32} trial values Y for the message block M_n, he computes that value $H'_{n-1}[Y] = D_Y(H_n)$ which H_{n-1} would need to have in order for H_n to have its correct value, under the assumption that Y were chosen for M_n. Each of these values is compared against the sorted

H.C. Williams (Ed.): Advances in Cryptology - CRYPTO '85, LNCS 218, pp. 14-17, 1986.

values for $H_{n-1}[X]$. If a match is found ($H_{n-1}[X] = H'_{n-1}[Y]$) then the assignments $M_{n-1} = X$, $M_n = Y$ complete the message M to one satisfying our requirements. Finally, the expected number of "successful" pairs (X,Y) is 1, so that we will find one with reasonable probability; this probability can be increased by a modest increase in the work factor.

The Davies-Price Scheme

Davies and Price [DP] introduced another DES-based message authentication scheme, by which they hoped to avoid this attack. Their scheme differs from Rabin's in that they cycle through the message blocks twice. Thus,

$$\begin{aligned}
&H_0 = \text{random}\\
&H_i = E_{M_i}(H_{i-1}), \qquad 1 \le i \le n\\
&H_{n+i} = E_{M_{n+i}}(H_{n+i-1}), \qquad 1 \le i \le n\\
&\text{RSA-Sign}\,(H_0, H_{2n}).
\end{aligned}$$

In the present note, we mount an attack on this scheme, similar to the meet-in-the-middle attack described above, with not much larger computational requirements.

The Attack

Our attack has two phases: a precomputation phase, which can be done once and used against all messages; and a stage tailored to the individual message. The requirements: for the precomputation stage, 2^{37} encipherments and 2^{36} storage; for the individual message, 2^{35} encipherments and 2^{32} storage. There are modest trade-offs available.

The message format is as follows. We select most of the message (say blocks M_{19} through M_n) to be the text of that bogus message which we are trying to authenticate. Blocks M_1 and M_2 are chosen (by a meet-in-the-middle step) to put ourselves into a standardized position. Finally, blocks $M_3, M_4, \ldots, M_{18}$ are chosen, from among possibilities enumerated during the precomputation, to "meet in the middle" one last time.

During the precomputation, we select an arbitrary 64-bit quantity Z, which is going be the value of $H_2, H_4, H_6, \ldots$, and H_{18}. We select 2^{36} trial values X, compute the values $E_X(Z)$, and sort and store these values. Now select 2^{36} trial values Y, compute the values $D_Y(Z)$, and compare each against the values $E_X(Z)$. Record each match: $E_X(Z) = D_Y(Z)$. We expect to find about 256 such pairs $\{(X_i, Y_i),\ 1 \le i \le 256\}$; if not, examine a few more values of Y. Each such pair (X_i, Y_i) can be used as a message pair (M_3, M_4), (M_5, M_6), ..., or (M_{17}, M_{18}), in the sense that if we have $H_2 = Z$, $M_3 = X_i$, $M_4 = Y_i$, then we get $H_4 = Z$.

Given a message $M' = (M_{19}, M_{20}, \ldots, M_n)$, an RSA-signature of some pair (H_0, H_{2n}), the chosen value of Z and the 256 pairs (X_i, Y_i) gotten during precomputation, our task is to select values of $M_1, M_2, \ldots, M_{18}$ which will make (H_0, H_{2n}) a valid hash of $M = (M_1, M_2, \ldots, M_n)$.

First we find values of M_1 and M_2 such that $E_{M_1}(H_0) = D_{M_2}(Z)$; this takes 2^{33} encipherments and 2^{32} storage. We know that $H_2 = Z$, so as long as the pairs $(M_3, M_4), \ldots, (M_{17}, M_{18})$ are chosen from our list (X_i, Y_i), we will have $H_4 = H_6 = \cdots = H_{18} = Z$. Assuming $H_{18} = Z$, use the values M_{19} through M_n to compute the value H_n; with the values M_1 and M_2 we can then get the value of H_{n+2}. Working backwards from H_{2n}, using the values $M_n, M_{n-1}, \dots, M_{19}$, we find the value of H_{n+18}.

Now we use the precomputed pairs (X_i, Y_i). For each of $256^4 = 2^{32}$ choices of four pairs (X_i, Y_i) to be the values of (M_3, M_4), (M_5, M_6), (M_7, M_8), (M_9, M_{10}), compute the value of H_{n+10} that would result. (The efficient way to do this is to run through the pairs lexicographically, so as not to recompute $E_{X_i}(H_{n+2})$ for each of 2^{24} occurrences.) Sort and store these trial values of H_{n+10}. Similarly, select pairs to be the values of (M_{11}, M_{12}), (M_{13}, M_{14}), (M_{15}, M_{16}), (M_{17}, M_{18}), and compute backwards from H_{n+18} to get trial values of H_{n+10}. Compare against the stored trial values. We expect one match, and the corresponding values of M_3 through M_{18} finish our task.

Extensions

The Davies-Price scheme could be altered by running through the message three times instead of twice. This attack will still work, at the expense of a large increase in the number of "constrained" message blocks (the message blocks chosen by the algorithm, rather than selected by the user).

Another possible scheme would be to set up two initializing vectors.

$$\begin{aligned}
&H_0,\quad H'_0 = \text{random}\\
&H_i = E_{M_i}(H_{i-1}),\quad 1 \le i \le n\\
&H'_i = E_{M_i}(H'_{i-1}),\quad 1 \le i \le n\\
&\text{RSA-Sign}\,(H_0, H_n, H'_0, H'_n).
\end{aligned}$$

Minor modifications to the present attack allow this scheme to be broken as well. Namely, do the same precomputation as before, and compute M_1, M_2 as before. Work forwards to find H_{n-2}, then use a meet-in-the-middle step to discover values M_{n-1}, M_n which satisfy the requirement on H_n. Then the values M_3 through M_{18} can be selected as before (from the pairs (X_i, Y_i)) to satisfy the requirements on H'_n.

A word about "constrained" message blocks: since we only need to examine $2^{36} < 23^8$ values X in the precomputation, we can select them to be EBCDIC representations of alphanumeric characters, so that even the "constrained" message blocks needn't look like total nonsense. In fact, at the risk of increasing the number of such blocks, we can increase their plausibility, to the point of having a set English text, with the freedom of choice made by substitution of synonyms. [DP]

Trade-offs

The presentation here tried to minimize computation time. There are two trade-offs available, which increase the computation time but decrease (1) storage and (2) length of constrained message, respectively.

When running a meet-in-the-middle attack, we work forward with J values, sort and store the outcomes; work backwards with K values, and compare against the J values stored. We are likely to succeed if $JK \ge N$, where N is the size of the space (in our case 2^{64}). Thus we trade off storage of J against computation time of K , subject to $K \ge J$, $JK \ge N$.

In the present attack, we had 18 blocks of constrained message. This can be decreased if we are willing to spend more time in precomputation. A precomputation of 2^{41} encipherments and 2^{40} temporary storage would allow us to recover $2^{16} = 65536$ pairs (X_i, Y_i), and with that larger selection we would need to add only ten constrained message blocks: two at the beginning as before, and four pairs (M_3, M_4), ... ,(M_9, M_{10}) to allow the last meet-in-the-middle step to go through ($65536^4 = 2^{64} = N$.)

References

[Den] D.E. Denning, Protecting public keys and signature keys, IEEE *Computer*, 1983, 16(2):27.

[DP] D.W. Davies and W.L. Price, "The Application of Digital Signatures based on Public Key Cryptosystems," NPL Report DNACS 39/80, National Physical Laboratory, Teddington, Middlesex, England, Dec. 1980.

[Rab] M. Rabin, Digital Signatures, in "Foundations of Secure Computation," Academic Press, New York, 1978.

[RSA] R.L. Rivest, A. Shamir, and L. Adleman, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems," Comm. ACM, vol. 21, no. 2, Feb. 1978, pp. 120-126.