1982

Signature Protocols for RSA and Other Public- Key Cryptosystems

Dorothy E. Denning

Report Number:

81-419

Signature Protocols for RSA and Other Public-Key Cryptosystems[1]

Dorothy E. Denning[2]

Computer Sciences Dept.
Purdue University
W. Lafayette, IN 47907

CSD-TR-419

October 1982
Revised: November 1982

Abstract. Public-key signature systems can be vulnerable to attack if the protocols for signing messages allow a cryptanalyst to obtain signatures on arbitrary messages of the cryptanalyst's choice. This vulnerability is shown to arise from the homomorphic structure of public-key systems. A signature protocol that foils the attack is described.

CR Categories and Subject Descriptors: E.3 [Data]: Data Encryption - public key cryptosystems.

General Term: Security.

Additional Key Words and Phrases: digital signature, cryptanalysis, cryptographic protocol, hashing, homomorphism.

1. Formerly "A Note on Strengthening RSA and Other Public-Key Cryptosystems".

2. Research supported in part by NSF Grant MCS80-15484.

1. Introduction

George Davida [1] has recently uncovered a potentially serious weakness in the basic protocol for signing messages using the RSA public-key cryptosystem [10]. Assuming that a cryptanalyst can get a user to sign arbitrary messages that may be meaningless, the cryptanalyst can decrypt ciphertext encrypted under the victim's public key or forge the victim's signature on a meaningful message. This is done by getting the victim to sign new messages derived from the intercepted ciphertext or chosen message. Although Davida refers to the attack as a "chosen signature" attack, it is actually a "chosen message" attack since the cryptanalyst chooses messages to be signed rather than signatures to be validated. The attack also works with other public-key systems.

The attack does not break the RSA system in the traditional sense whereby a cryptanalyst can obtain secret keys. Indeed, the attack is carried out without knowledge of the victim's private key. In this sense, the attack is much weaker than a "chosen plaintext" attack on a conventional cryptosystem, which, if successful, breaks the system.

We believe that a signature protocol for public-key systems developed by Davies and Price [2] foils the attack for any public-key system. We shall first describe the weakness in the basic RSA protocol, and then show how it generalizes to other public-key systems. We shall then describe the protocol by Davies and Price.

2. The Basic RSA Protocol

Let n be the modulus for the victim's RSA cryptosystem, where $n = pq$ for large secret primes p and q; and let e and d be the public and private exponents respectively, where e and d are multiplicative inverses mod $\varphi(n) = (p-1)(q-1)$. The public exponent e is used to encipher and validate signatures; the private exponent d to decipher and sign messages. To send the user a secret message M, the sender enciphers M by computing $C = M^e \bmod n$; the user deciphers the ciphertext C by computing $C^d \bmod n = M$. Similarly, the user signs a message M by computing $S = M^d \bmod n$; the receiver validates the signature S by computing $S^e \bmod n = M$. The security of the system rests on the assumption that a cryptanalyst cannot determine the factors p and q of n. (See [5] for a tutorial on the number theory behind the RSA system.)

3. The Potential Weakness

Suppose that a cryptanalyst has intercepted ciphertext C sent to the victim, where $C = M^e \bmod n$. Davida [1] has shown how the cryptanalyst may be able to determine M without knowing the deciphering exponent d. His method works as follows:

Algorithm 1. Davida's method for obtaining $C^d \bmod n = M$.

1. Factor C into $t \geq 2$ components, obtaining $C = C_1 C_2 \cdots C_t$ (the components C_i need not be prime or prime powers -- any decomposition of C will do). This implies that M also factors into t corresponding components $M_1, \ldots, M_t$, where

   $$C = C_1 C_2 \cdots C_t = (M_1 M_2 \cdots M_t)^e \bmod n = (M_1)^e (M_2)^e \cdots (M_t)^e \bmod n \, ,$$
   and $M_i = C_i^d \bmod n \quad (i = 1, \ldots, t)$.

2. Get the victim to sign a message X and messages $XC_1 \bmod n, \ldots, XC_t \bmod n$. X can be a new message or a message previously signed by the victim. The messages $XC_1, \ldots, XC_t$ might be lines in some file the cryptanalyst requests the victim to sign line by line to acknowledge receipt. The signatures obtained are thus:

   $$S = X^d \bmod n$$
   $$S_i = (XC_i)^d \bmod n \quad (i = 1, \ldots, t) \, .$$

3. Compute the multiplicative inverse $S^{-1} \bmod n$ of the signature S, getting

   $$S^{-1} = X^{-d} \bmod n \, .$$

4. Multiply this by each of the S_i to obtain the M_i:

   $$S^{-1} S_i \bmod n = X^{-d} (XC_i)^d \bmod n = C_i^d \bmod n = M_i \, .$$

5. Compute the product $M_1 M_2 \cdots M_t \bmod n = M$.

The attack can also be made using $X = 1$. Then the cryptanalyst needs only the t signatures $S_i = C_i^d \bmod n = M_i$. This attack, however, may be easier to detect since the factors of C and M are exposed.

In the unlikely event that S is not relatively prime to n, S does not have a unique inverse mod n. But in this case, the cryptanalyst can factor n because S will be a multiple of p or q, whence $gcd(S, n) = p$ or q.

Judy Moore has uncovered an even simpler attack that requires only one signature:

Algorithm 2. Moore's method for obtaining $C^d \bmod n = M$.

1. Pick an arbitrary S and compute $X = S^e \bmod n$; this implies

$$S = X^d \bmod n \ .$$

2. Get the victim to sign the message $XC \bmod n$, obtaining the signature

$$S_1 = (XC)^d \bmod n \ .$$

3. Compute $S^{-1} \bmod n = X^{-d} \bmod n$.

4. Multiply S^{-1} by S_1 to obtain M:

$$S^{-1}S_1 \bmod n = X^{-d}(XC)^d \bmod n = C^d \bmod n = M \ .$$

Under the assumption that the RSA system is cryptographically strong (computationally infeasible to break), Moore's method is optimal in the sense of requiring the minimal number of signatures from the victim. If it were not, then we could decrypt ciphertext without any cooperation from the victim, thereby breaking RSA.

Because both algorithms compute $C^d \bmod n$, they may also be used to forge the victim's signature on a message C chosen by the cryptanalyst.

4. Generalizing the Results to Other Public-Key Systems

Consider an arbitrary public-key cryptosystem with private deciphering (signature) transformation D and public enciphering transformation $E = D^{-1}$. We initially assume the public-key system can be used for both message encryption (i.e., $D(E(X)) = X$ can be computed) and signatures (i.e., $E(D(X)) = X$ can be computed); signature-only systems are considered later.

Moore's method extends to the system if the message space forms a group with binary operator $\circ$ and identity element 1; the signature space forms a group with a binary operator, also denoted by $\bullet$, and identity 1; and the deciphering transformation D is a homomorphism from the message group to the signature group; that is, the following properties hold for all messages X, Y, and Z:

1. $X \bullet (Y \bullet Z) = (X \bullet Y) \bullet Z$ (Associativity - for Step 4)
2. $X \bullet 1 = 1 \bullet X = X$ (Identity - for Steps 3 and 4)
3. $X \bullet X^{-1} = X^{-1} \bullet X = 1$ (Inverses - for Step 3)
4. $D(X \bullet Y) = D(X) \bullet D(Y)$ (Homomorphism - for Step 4)

Algorithm 3 computes $D(C)$ to decrypt C or forge the victim's signature on C:

Algorithm 3. Generalization of Moore's method to obtain $D(C)$.

1. Pick S and compute $X = E(S)$; this implies $S = D(X)$.
2. Get the victim to sign the message $X \bullet C$. The signature is $S_1 = D(X \bullet C)$.
3. Compute S^{-1}.
4. Compute

$$\begin{aligned} S^{-1} \bullet S_1 &= S^{-1} \bullet D(X \bullet C) = S^{-1} \bullet (D(X) \bullet D(C)) \\ &= S^{-1} \bullet (S \bullet D(C)) = (S^{-1} \bullet S) \bullet D(C) \\ &= 1 \bullet D(C) = D(C) . \end{aligned}$$

The RSA system fits this general pattern, where both the message and signature groups are defined by the integers relatively prime to n together with multiplication. The deciphering transformation is a homomorphism because

$$(XY)^d \bmod n = [(X^d \bmod n)(Y^d \bmod n)] \bmod n .$$

It is not surprising that a cryptosystem for which the deciphering transformation is a homomorphism is vulnerable to certain types of attack. Rivest, Adleman, and Dertouzos [9] showed that such cryptosytems can have inherent weaknesses.

We now consider the case where the public-key system is a signature-only system. Thus, the cryptanalyst is interested only in forging a signature on a message C. To see how Moore's method can be applied in this case, we consider the individual steps in Algorithm 3. Step 1 cannot be performed because messages cannot be enciphered in a signature-only system. But since the objective is to obtain X and S such that $S = D(X)$, this step can be replaced by one that obtains the victim's signature on a message X picked by the cryptanalyst. Steps 2 and 3 can be performed without modification. Step 4 can be performed as long as D is a homomorphism. If D is not a homomorphism but E is, then a slightly different approach can be taken in Step 4 since the objective is simply to find a signature that passes the validation test. Algorithm 4, which generalizes a method by DeMillo and Merritt [4], uses this approach:

Algorithm 4. Forge a Signature on C.

1. Pick X and get the victim to return a signature $S = D(X)$; this implies $X = E(S)$.
2. Get the victim to return the signature $S_1 = D(X \cdot C)$.
3. Compute S^{-1}.
4. Compute the signature $S_2 = S^{-1} \cdot S_1$. S_2 is a valid signature of C because

$$\begin{aligned} E(S_2) &= E(S^{-1} \cdot S_1) = E(S^{-1}) \cdot E(S_1) \\ &= E(S)^{-1} \cdot E(S_1) \qquad \text{(because } E \text{ is a homomorphism)} \\ &= X^{-1} \cdot (X \cdot C) = C \; . \end{aligned}$$

Shamir's signature-only knapsack scheme [11] (see also [5]) fits this pattern. Here, the signature validation transformation E is given by $E(S) = SA \bmod n$, where n is a k-bit prime, A and S are integer vectors of length $2k$, and SA denotes the scalar product. where X is $\overset{an}{\cancel{a}}$ integer $\bmod n$. The message group is defined by the integers $\bmod n$ with addition and identity 0;

the signature group by integer vectors of length $2k$ with vector addition and identity $\underline{0}$. Although D is not a homomorphism, E is a homomorphism from the signature group to the message group, since for all signatures S_1 and S_2:

$$(S_1 + S_2)A \bmod n = (S_1A \bmod n + S_2A \bmod n) \bmod n \ .$$

For Shamir's system, Algorithm 4 becomes the method in DeMillo and Merritt [4]:

Algorithm 5. DeMillo's and Merritt's Method for Forging a Signature S_2 on C with Shamir's Signature System.

1. Pick X and get the victim to return the signature S such that $SA \bmod n = X$.
2. Get the victim to return a signature S_1 on $X + C$ such that $S_1A \bmod n = X+C$.
3. Compute $S^{-1} = -S$.
4. Compute the signature $S_2 = -S + S_1$. S_2 is a valid signature of C because

$$\begin{aligned} E(S_2) &= E(-S + S_1) = (-S + S_1)A \bmod n \\ &= (-SA \bmod n + S_1A \bmod n) \bmod n = [-X + (X + C)] \bmod n = C \ . \end{aligned}$$

DeMillo and Merritt also consider similar attacks on variants of the RSA system. These systems all have a underlying homomorphic structure (though not explicitly identified as such in their paper), which explains their vulnerability to this general method of attack.

5. An Improved Protocol

For the attacks to succeed, the cryptanalyst must be able to get the victim to sign essentially arbitrary messages that are supplied by the cryptanalyst and are not likely to be meaningful. To protect against such attacks, users can sign only meaningful messages of their choice. Messages received from other users can be modified before signing.

We now describe a protocol that protects against the attacks by transforming messages with a one-way public function h before signing. A message M is thus signed by computing

$$S = D(h(M)) .$$

The function h should satisfy four properties:

1. h should destroy all homomorphic structure in the underlying public-key cryptosystem; that is, $h(X \cdot Y) \neq h(X) \cdot h(Y)$ must hold. Moreover, for almost all X and Y, $D(h(X \cdot Y)) \neq D(h(X)) \cdot D(h(Y))$ should hold. Then the cryptanalyst cannot factor out the X or Y in a signature $D(h(X \cdot Y))$.
2. h should be computed over entire messages (rather than on a block by block basis). This will make it difficult for a cryptanalyst to obtain signatures by inserting blocks into a file that otherwise looks legitimate.
3. h should be one-way so that the cryptanalyst cannot obtain a signature on a message X by requesting a signature for $h^{-1}(X)$.
4. h should have the property that for any given message X and value $h(X)$, it is computationally infeasible to find another message Y such that $h(Y) = h(X)$. This is needed to prevent forgeries since $h(X)$ can be computed from a signature $S = D(h(X))$ by applying the public function E to S.

Applying the transformation h to a message before signing has an additional benefit. Because the transformations $E(\cdot)$ and $D(h(\cdot))$ are not inverses, a cryptanalyst cannot hope to decrypt an intercepted ciphertext message C by getting a signature on C.

The idea of transforming messages before signing is due to Davies and Price [2], who designed a protocol for signing secret messages using a one-way public hashing function h that conceals messages and prevents forgeries. Their function h blocks a message M into 56-bit blocks $M_1, \ldots, M_r$ and computes a digest

$$\overline{M} = h(M,I) = E_{M_r} \circ \cdots \circ E_{M_1} \circ E_{M_r} \circ \cdots \circ E_{M_1}(I) ,$$

where E_{M_i} is the DES enciphering algorithm keyed to block M_i, I is a random 64-bit initialization seed for the DES, and "o" denotes function composition. Because the function h can be computed both forwards and backwards (by using the deciphering transformations D_{M_i}) for an arbitrary message M, the message must be repeated in the keys to prevent a "meet in the middle" forgery (compute forwards from I using 2^{32} variations of the first half of the desired message and backwards from $\overline{M}$ using 2^{32} variations of the second half of the message; sort the results to find a match, which is likely to occur for "birthday problem" reasons).

Wolfgang Bitzer has suggested an improved hashing function that foils the meet in the middle attack with a single pass over the message. His hashing function is given by

$$\overline{M} = h(M,I) = Z_{r+1} ,$$

where

$$Z_{i+1} = E_{Z_i' \oplus M_i}(Z_i) \quad (1 \le i \le r)$$
$$Z_1 = I,$$

Z_i' consists of 56 bits selected from Z_i, and $\oplus$ denotes exclusive or. The meet in the middle attack is prevented because it is not possible to compute backwards through the function (i.e., compute Z_i from Z_{i+1}).

Both DES-based hashing functions would destroy the multiplicative structure of the RSA system and the additive structure of knapsack systems. They almost certainly would destroy the homomorphic structure of any underlying cryptosystem.

Using the hashing function, the message M is signed as $S = D([\overline{M}, I])$, where the 128-bit block $[\overline{M}, I]$ is replicated as many times as necessary to fill the input block for the signature transformation D.

A signature S on an alleged message M is validated by first computing $E(S) = D^{-1}(S) = [\overline{M}, I]$; next computing $h(M,I)$ using the public function h and the alleged message M; and finally comparing $h(M,I)$ with $\overline{M}$. We conjecture, but have not been able to prove, that a constant seed I_0 could be used for all messages without compromising security (in the same way that a constant seed is used for one-way encryption of passwords). Then the signature would be simply $S = D(h(M))$ as suggested earlier.

The message M can be transmitted either as cleartext (if secrecy is not needed), as ciphertext encrypted using the receiver's public key, or as ciphertext encrypted using a secret key shared by the sender and receiver (if a conventional cryptosystem is used for message secrecy, with the public-key system reserved for signatures and key exchange).

The hashing function has two important advantages besides protecting against signature attacks

1. It separates the signature transformation from the secrecy transformation, allowing secrecy to implemented with a one-key system or to be skipped [2]. Yet the separation is achieved without much message expansion, since each signature is a single block.

2. It conceals messages so that signatures can be publicly disclosed without revealing their corresponding messages. This is important for recovering from compromises or direct disclosure of private keys. Let D_A be the signature key of user A. In order that a signature S of A can remain valid after D_A is compromised or deliberately disclosed, S must be bound to A's current public key E_A, timestamped, and signed by the public key server (notary public) [8], giving a "signature certificate"

[6] $G = D_P(T, A, E_A, S)$, where D_P is the signature key of the public key server, and T is the timestamp. And in order that S can remain valid even if D_P is compromised, G must be kept in a public log. For this reason, it is important that S conceal the message signed and have minimal storage requirements. This is achieved with the hashed signature method. (For further discussion of this, see [6].)

6. Conclusions

Davida's discovery demonstrates the fundamental importance of encryption protocols. It is not enough to have an encryption algorithm that is computationally hard to break; the protocols for using the algorithm must also withstand attack. We have identified several properties that should be satisfied by any signature protocol; in particular, it should destroy any homomorphic structure in the underlying public-key algorithm. The signature protocol described here appears to satisfy these properties. Further research along the lines initiated by Dolev and Yao [7] and DeMillo, Lynch, and Merritt [3] is needed for proving the security of protocols.

Acknowledgements

This note evolved from discussions with many people. When George Davida told me of his attack on RSA, I wrote a preliminary note explaining how the attack could be foiled with hashing. When Judy Moore told me of her simple attack, and Michael Merritt told me of his results with DeMillo to extend Davida's attack to other cryptosystems, I began to search for the common mathematical structure of the attacks. Independently, DeMillo and Merritt identified a similar mathematical structure. Many others have provided helpful suggestions, including Bob Blakley, Peter Denning, Ron Graham, Ron Rivest, and the students in my cryptography class. Donald Davies brought Bitzer's hashing function to

my attention.

References

1. Davida, G. I., "Chosen Signature Cryptanalysis of the RSA (MIT) Public Key Cryptosystem," TR-CS-82-2, Dept. of EE and CS, Univ. of Wisconsin, Milwaukee, WI (October, 1982).

2. Davies, D. W. and Price, W. L., "The Application of Digital Signatures Based on Public Key Cryptosystems," NPL Report DNACS 39/80, National Physical Lab., Teddington, Middlesex, England (Dec. 1980).

3. DeMillo, R., Lynch, N. A., and Merritt, M. J., "Cryptographic Protocols," *Proc. SIGACT Conf.*, (1982).

4. DeMillo, R. and Merritt, M. J., "Chosen Signature Analysis of Public-Key Cryptosystems," Technical Memorandum, Georgia Tech. (Oct. 1982).

5. Denning, D. E., *Cryptography and Data Security*, Addison-Wesley, Reading, Mass. (1982).

6. Denning, D. E., "Protecting Public Keys and Signature Keys," *IEEE Computer*, (to appear 1983).

7. Dolev, D. and Yao, A. C., "On the Security of Public Key Protocols," *Proc. 22nd Annual Symp. on the Foundations of Computer Science*, (1981).

8. Merkle, R. C., "Protocols for Public Key Cryptosystems," pp. 122-133 in *Proc. 1980 Symp. on Security and Privacy*, IEEE Computer Society (April 1980).

9. Rivest, R. L., Adleman, L., and Dertouzos, M. L., "On Data Banks and Privacy Homomorphisms," pp. 169-179 in *Foundations of Secure Computation*, ed. R. A. DeMillo et al.,Academic Press, New York (1978).

10. Rivest, R. L., Shamir, A., and Adleman, L., "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems," *Comm. ACM* 21(2) pp. 120-126 (Feb. 1978).

11. Shamir, A., "A Fast Signature Scheme," MIT/LCS/TM-107, MIT Lab. for Comp. Sci., Cambridge, Mass. (July 1978).