J. Cryptology (1993) 6: 55–61

Journal of Cryptology

Cascade Ciphers: The Importance of Being First*

Ueli M. Maurer
Institute for Theoretical Computer Science, ETH Zürich,
CH-8092 Zürich, Switzerland

James L. Massey
Institute for Signal and Information Processing, ETH Zürich,
CH-8092 Zürich, Switzerland

Communicated by Ernest F. Brickell

Received 17 July 1990 and revised 5 May 1992

Abstract. The security of cascade ciphers, in which by definition the keys of the component ciphers are independent, is considered. It is shown by a counterexample that the intuitive result, formally stated and proved in the literature, that a cascade is at least as strong as the strongest component cipher, requires the uninterestingly restrictive assumption that the enemy cannot exploit information about the plaintext statistics. It is proved, for very general notions of breaking a cipher and of problem difficulty, that a cascade is at least as difficult to break as the first component cipher. A consequence of this result is that if the ciphers commute, then a cascade is at least as difficult to break as the most-difficult-to-break component cipher, i.e., the intuition that a cryptographic chain is at least as strong as its strongest link is then provably correct. It is noted that additive stream ciphers do commute, and this fact is used to suggest a strategy for designing secure practical ciphers. Other applications in cryptology are given of the arguments used to prove the cascade cipher result.

Key words. Cascade ciphers, Multiple encryption, Provable security, Computational security.

1. Introduction

An important general question in cryptography, which has, for instance, been addressed (but not answered) by Diffie and Hellman [2, p. 83] and Merkle and Hellman [5], is whether multiple encryption with a certain cipher increases its cryptographic security. In this paper we consider the more general question of the

* The results of this paper were presented in part at the 1990 IEEE Symposium on Information Theory, January 14–19, 1990, San Diego, California.

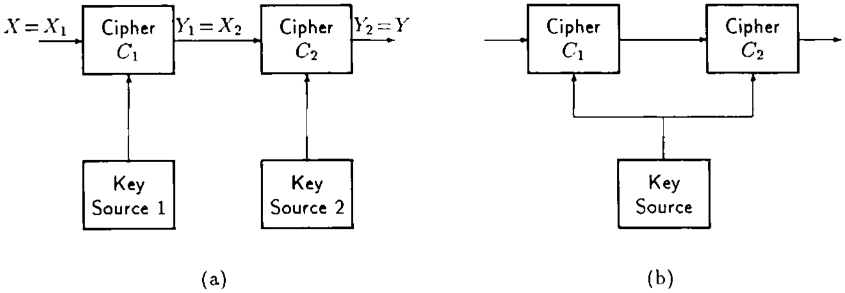

Fig. 1. A cascade of two ciphers (a) and a product cipher consisting of two ciphers (b). The secret keys within a cascade cipher are statistically independent but the subkeys of a product cipher need not be independent.

security of cascade ciphers where the component ciphers can be distinct. Can, for example, DES be weakened by cascading it with another cipher?

Obviously, if a binary data sequence is encrypted twice with the same binary additive stream cipher and if the key happened to be the same for both encryptions, then the resulting ciphertext is equal to the plaintext and thus the system is completely insecure. Such considerations have caused some cryptographers to worry about the security of multiple encryption.

The distinction between *cascade ciphers* and *product ciphers* [6] is that in the latter the keys of the component ciphers need not be statistically independent, whereas they are in the former. This difference is illustrated in Fig. 1. Note that the above stream-cipher example illustrates the impossibility of proving a general result about the security of product ciphers.

We consider the two most popular types of ciphers, block ciphers and additive stream ciphers, but the results can easily be generalized for arbitrary ciphers with compatible input and output alphabets. Throughout this paper we consider the difficulty of a problem to be its *intrinsic* difficulty as opposed to the (often considered) *historical* difficulty, which is defined as the difficulty when the optimum *known* algorithm is used. For a given model of computation, the intrinsic difficulty of a problem is constant (though usually unknown), but the historical difficulty may decrease as new algorithms are discovered.

It seems to be intuitively clear, and in fact the design of some practical ciphers has been motivated by this idea, that if one of the ciphers in a cascade is strong, then the whole cascade is strong. On the other hand, it seems to be at least conceivable that a cipher can be weakened by cascading it with some malignant cipher especially designed for this purpose. The following theorem for block ciphers, which was proved by Even and Goldreich [3], is therefore nontrivial. However, we shall see that it holds only under the uninterestingly restrictive assumption, which is not explicitly stated in [3], that the enemy cannot exploit information about the plaintext statistics. The history of cryptology shows that most successful attacks on ciphers have succeeded precisely because they exploited knowledge of the plaintext statistics.

For emphasis, we repeat here our standing assumption that *the keys of the component ciphers in a cascade are selected statistically independently.*

Folk Theorem. *A cascade of ciphers is at least as difficult to break as any of its component ciphers.*

The following plausibility argument can be used for "proving" this theorem. It suffices to prove the theorem for a cascade of two ciphers as the general case then follows by a simple induction. Assume there exists an efficient algorithm, Algorithm A, that breaks the cascade of ciphers C_1 and C_2 (see Fig. 1(a)), i.e., that determines the plaintext X corresponding to a given cryptogram Y of the cascade cipher, when some pairs of corresponding plaintext and ciphertext blocks (for the same key) are given. (This corresponds to a so-called known-plaintext attack, but the arguments below are easily modified for a chosen-plaintext or a chosen-ciphertext attack.) It remains to show that Algorithm A can be used to break both component ciphers, i.e., to determine the plaintext for a given cryptogram for either one of these ciphers when some plaintext/cryptogram pairs for this cipher are given. Suppose now that the second component cipher C_2 is to be attacked. Given the cryptogram Y_2 and some plaintext/ciphertext pairs for C_2, we can attack C_2 by considering it to be embedded in a cascade in which we ourselves choose the key of the first cipher C_1 in the cascade. Because we know the key for cipher C_1 in this artificial cascade, we can convert each known plaintext/ciphertext pair for C_2 to a plaintext/ciphertext pair for the cascade simply by decrypting the given plaintext considered as a ciphertext for C_1. Here, we make the reasonable assumption that the difficulty of encrypting or decrypting for a component cipher with known key is negligible compared with the difficulty of breaking that cipher. Now we apply Algorithm A to the artificial cascade and thus determine the plaintext X corresponding to the ciphertext $Y = Y_2$ of the cascade. Finally, we encrypt X with cipher C_1 to obtain the desired plaintext X_2 of the cipher C_2 under attack. An entirely parallel argument shows that Algorithm A can also be used to break the first component cipher C_2 with essentially the same amount of computation as required for Algorithm A to break the cascade.

Essentially the same argument as above applies if "breaking the cipher" means determining the key. When the breaking algorithm applied to the artificial cascade yields the keys of the two component ciphers, we simply accept the key of the component cipher actually under attack.

It is important to note that the above plausibility argument, which is essentially the proof used by Even and Goldreich [3], is valid only for "pure" known-plaintext, chosen-plaintext, or chosen-ciphertext attacks in which *the enemy cannot make use of information about the statistics of the plaintext to be found.* It is therefore not even valid for a ciphertext-only attack in which only information about the plaintext statistics is exploited.

To see that the plaintext statistics are crucial, consider the following counterexample to the above "folk theorem." Consider two block ciphers, C_1 and C_2, which both have input and output alphabet $\{A, B, C, D\}$ and key space $\{0, 1\}$ with 0 and

1 being equiprobable. The block size is one digit, and the key is used one time. For keys 0 and 1, cipher C_1 transforms (A, B, C, D) into (C, D, A, B) and (C, D, B, A), respectively, and cipher C_2 transforms (A, B, C, D) into (C, D, A, B) and (D, C, A, B), respectively. Assume that the plaintext source statistics are such that it emits only A or B with nonzero probability. Then C_1 is completely insecure for this source, but C_2 is perfectly secure since the plaintext and ciphertext are statistically independent. However, the cascade with C_1 preceding C_2 is completely insecure because A and B are transformed into C and D, respectively, by C_1, then back into an A and B, respectively, by C_2. Hence the cascade cipher, which for a source emitting only A's and B's is equivalent to the identity transformation, is much weaker than its component cipher C_2, which contradicts the folk theorem.

2. Security of Cascade Ciphers

In the following we make the usual assumption that the enemy knows precisely the cipher system, including the probability distribution of the key, but that he has no other direct *a priori* information about the key. We further allow that the enemy's knowledge, in addition to complete knowledge of the ciphertext, is a subset of the following:

(1) Complete or partial knowledge of the plaintext statistics.
(2) For block ciphers: (a) the corresponding ciphertext blocks for some chosen and/or known plaintext blocks and/or (b) some information about the plaintext blocks corresponding to some chosen ciphertext blocks for the same key.
(3) For additive stream ciphers, (possibly partial) knowledge of some portion of the keystream.

It should be noted that assumption (2) for block ciphers is not completely general since it specifies that ciphertext must be available in entire encipherable/decipherable units, i.e., as complete blocks. Additive stream ciphers, for which known-plaintext, chosen-plaintext, and chosen-ciphertext attacks are all equivalent to knowledge of a portion of the keystream, do not require a similar restriction of generality. The following proposition holds for any attack covered by the above general assumption, and for virtually any reasonable definition of breaking a cipher.

Proposition. *A cascade of n ciphers is at least as difficult to break as the first cipher in the cascade (under the reasonable assumption that the difficulty of carrying out k encryption or decryption operations is negligible compared with the difficulty of breaking the cascade, where k is the number of plaintext or ciphertext units used in the attack).*

Proof. Assume an oracle who gives upon request and free of cost the keys of all component ciphers in the cascade except the key of the first component cipher. Breaking the cascade with the oracle's help cannot be more difficult than breaking it without this help because the oracle's information can always be disregarded. However, breaking the cascade with the oracle's help is equivalent to breaking the first cipher with the oracle's help because every cryptogram of the cascade can with assumed negligible computation be converted into the corresponding cryptogram for the first cipher and vice versa and because the plaintexts of the first cipher and the cascade are the same. However, breaking only the first cipher with the oracle's help is equivalent to breaking this first cipher without the oracle's help. This follows from the fact that the information provided by the oracle is statistically independent of the first key. In other words, it follows from the fact that if the cryptanalyst attacking the first cipher wishes to embed that cipher in an artificial cascade in which he himself chooses the second and all subsequent keys (that by our standing assumption for a cascade are independent of the first key) so as to avail himself of the oracle's aid, then he already possesses all the information that the oracle can provide. It follows that breaking the first cipher in the cascade cannot be more difficult than breaking the cascade cipher itself. □

Remark. It should be noted that the proof of [3] for the Folk Theorem, which was in the previous section argued to hold only under strong restrictions, is a valid alternative proof for the above proposition.

When the component ciphers in a cascade commute, i.e., when the enciphering transformation of the cascade is independent of the order of the component ciphers, then every cipher can be considered as being first.

Corollary 1. *A cascade of commuting ciphers is at least as difficult to break as the most-difficult-to-break component cipher.*

We remark that additive stream ciphers, i.e., ciphers in which a key-dependent "keystream" sequence is added bit-by-bit modulo 2 to the plaintext sequence, do commute. With respect to provable security of cascade ciphers, additive stream ciphers seem to be superior to block ciphers for two reasons: first, because they commute and thus Corollary 1 applies, and, second, because the smallest encipherable/decipherable unit is a single bit rather than a block of bits, and therefore our assumption (needed for provable security) that ciphertext must be available in an attack as complete transformable units entails no loss of generality.

Corollary 2. *The bitwise modulo 2 sum of n keystream sequences that are generated by devices with independent keys is at least as difficult to predict as the most-difficult-to-predict keystream sequence.*

An analogous corollary could be formulated for the difficulty of distinguishing a pseudorandom generator with random seed from a binary symmetric source, a notion of security introduced by Yao [7] within a complexity-theoretic framework, or for any other reasonable notion of stream-cipher security.

3. Applications and Conclusions

Corollary 2 has implications for the practical design of additive stream ciphers. Since none of the known design methods for stream ciphers (or any other cipher for that matter) yields provably secure ciphers, it seems to be advisable to cascade a small set of keystream generators, each relying on a different design principle, rather than to employ one large keystream generator relying on only a single design principle. The cascade cipher can fail only if all applied design principles happen to fail simultaneously. For example, a cipher might be devised that could possibly be insecure only if factoring, the discrete logarithm, and other conjectured hard problems were all easy to solve. The cascade approach to founding cryptographic security on different conjectured hard problems seems to be safer than approaches such as the elegant schemes of [4] and [1] for proving that a specific key-distribution protocol and a specific interactive-identification scheme, respectively, are at least as secure as the factoring and the discrete logarithm problems are difficult. Naturally, in the cascade approach to stream-cipher design, the total cost and key size available must be divided among the component ciphers and the key size of every component cipher must be large enough (e.g., 100 bits) to make an exhaustive search over the key space infeasible. Note that although it can be proved "only" that a cryptographic stream-cipher chain is always at least as strong as the strongest link, it can be reasonably conjectured that the cascade is usually much stronger. Such a conjecture is comparable with other conjectures on which the security of practical ciphers relies.

Another implication of Corollary 2 is on the unsolved problem of finding provably (as opposed to conjectured) computationally secure ciphers.

Corollary 3. *A cascade of additive binary stream ciphers, known to contain at least one computationally secure cipher, is computationally secure.*

Note that Corollary 3 does not require that it be known which of the component ciphers is the computationally secure one. This suggests the new problem of proving that some set of ciphers must contain at least one computationally secure cipher.

The arguments used above to prove the cascade cipher result have other applications in cryptology. Consider the security of RSA public exponents. Is $e = 3$ or $e = 15$ more secure? It can be argued that an exponent e is at least as secure as any of its divisors (e.g., $e = 15$ is at least as secure as $e = 3$ or $e = 5$), since any algorithm extracting eth roots modulo m, where $e = te'$, can be used for extracting e'th roots by raising the input number to the power t. However, two remarks are in order. First, this argument applies only if the modulus is selected randomly from a large set of moduli, since, for any fixed modulus, there exists a fast (but possibly unknown) algorithm for extracting eth roots. Second, as in the case with secret-key block ciphers, this argument is valid only under the assumption that ciphertext is available as entire blocks.

Consider the security of a pseudorandom number generator (with a random seed) against being distinguished from a binary symmetric source (BSS), i.e., a device whose output is a true coin-tossing sequence. Applying an invertible and non-expanding transformation to the generator's output cannot decrease, but can possibly strongly increase, its security against such a distinguishability attack. A statistical test distinguishing the modified generator from a BSS can easily be converted into a statistical test distinguishing the original generator from a BSS with the same probability of success if we assume that the difficulty of making the transformation is negligible compared with the difficulty of the distinguishability attack on the pseudorandom number generator itself. Notice that we require the invertible transformation to be easily computable only in the forward direction. A (conjectured) one-way function (e.g., exponentiation modulo a prime) can thus be applied for increasing a generator's security without any risk of introducing a trapdoor for the enemy.

The reason for presenting this paper in a fairly informal manner is not that the results would not hold if all definitions were to be formalized precisely, but rather that the results hold for every reasonable formalism. For example, the difficulty of a problem could be measured as the average (over all instances of the problem) time required to solve it by the optimal program on a certain specific computer, and breaking an additive stream cipher could be defined as predicting the next bit, with probability $1/2 + \varepsilon$ of success for some given positive ε, when given a certain portion of the keystream, or as distinguishing the keystream generator from a binary symmetric source with a certain probability of success.

References

[1] E. F. Brickell and K. S. McCurley, An interactive identification scheme based on discrete logarithms and factoring, *Advances in Cryptology—Eurocrypt '90* (ed. I. B. Damgaard), pp. 63–71, Lecture Notes in Computer Science, Vol. 473, Berlin: Springer-Verlag, 1991.

[2] W. Diffie and M. E. Hellman, Exhaustive cryptanalysis of the NBS Data Encryption Standard, *IEEE Computer Magazine*, Vol. 10, No. 6, June 1977, pp. 74–84.

[3] S. Even and O. Goldreich, On the power of cascade ciphers, *ACM Transactions on Computer Systems*, Vol. 3, 1985, pp. 108–116.

[4] K. S. McCurley, A key distribution system equivalent to factoring, *Journal of Cryptology*, Vol. 1, No. 2, 1988, pp. 95–105.

[5] R. C. Merkle and M. E. Hellman, On the security of multiple encryption, *Communications of the ACM*, Vol. 24, No. 7, July 1981, pp. 465–467.

[6] C. E. Shannon, Communication theory of secrecy systems, *Bell System Technical Journal*, Vol. 28, Oct. 1949, pp. 656–715.

[7] A. C. Yao, Theory and applications of trapdoor functions, *Proceedings of the 23rd Annual Symposium on Foundations of Computer Science*, IEEE, New York, 1982, pp. 80–91.