

”Key” Issues in Cryptography

Henk C.A. van Tilborg
Dept. of Mathematics and Computing Science
Eindhoven University of Technology
P.O. Box 513
5600 MB Eindhoven, The Netherlands
E-mail: H.C.A.v.Tilborg@tue.nl

1 Introduction

All cryptographic systems, no matter if they are directly used for privacy or authentication purposes or if they are embedded in much more complex systems like electronic payment, make use of a key. In symmetric systems, like AES or message authentication codes, sender and receiver share a common key. In asymmetric systems, each user has two matching keys, one of which is public while the other is kept secret. When banking over the net, the customer may not even be aware of the underlying cryptography unless he notices the little padlock on the screen, but both symmetric and asymmetric systems are being used there. The security of these systems depends exclusively on the key, at least if the system is well-designed. To make cryptosystems practical, easy generation and handling of the keys is also crucial.

There are many important issues related to keys, most notably all the well-known problems around key management (generation, distribution, storage, and the entire life cycle). However, here we want to focus on two more mathematical aspects that in our view are and remain important in the future:

1. The generation of keys.
2. The length of keys.

Both aspects give rise to their own fundamental questions and practical issues.

With respect to the generation of keys, we see fundamental questions and practical issues to be resolved. A basic question is how to generate provably

secure keys in an efficient way. Another one is how to efficiently generate long key streams while still being able to prove something about their cryptographic strength.

With regards to the lengths of the keys that one uses, we note that increasing computer power is creating a growing gap between the key lengths of symmetric cryptosystems and of asymmetric cryptosystems. This is highly unfortunate, since asymmetric systems have many natural advantages, most importantly the inherent possibility to distinguish the legitimate user from other parties involved. Any theoretical or practical technique to reduce this gap in size is of utmost importance.

2 The Generation of Keys

The efficient generation of keys to be used for symmetric systems like AES is non-trivial. An approach is to start with a truly random sequence and expand it with a deterministic algorithm to a longer sequence with the property that a probabilistic, polynomial-time algorithm cannot distinguish it from a sequence that has a uniform distribution (the probability that it can successfully distinguish decreases faster than any polynomial in the length). The theory of (provably secure) pseudo-random generators is directly related to the theory of one-way functions. Unfortunately, there is not even a proof of the existence of one-way functions. All existing constructions depend on mathematical assumptions. Much work needs to be done here.

A recent idea is to use elliptic curves to construct pseudo-random generators. A parameter to play with is the number of bits that are generated per iteration step. In practical implementations, one may want to sacrifice the security for the benefit of speed and complexity.

The generation of long running keys for encryption purposes has been studied for over fifty years. It is still highly relevant in applications where large amounts of data have to be encrypted in real time (e.g. pay-tv). These systems are based on linear recurrence relations over finite fields. In cryptographic applications, linear systems cannot be used directly, so non-linearity is commonly used one way or another, but that makes their analysis right away impossible or very complex. New insights are necessary here.

There is some recent work on recurrence relations over elliptic curves for the generation of long pseudo-random sequences. At this moment it is not clear if this approach will lead to efficient high-speed generation of long running keys and if sufficiently strong statements can be proved about their cryptographic strength.

A meeting ground for the two sub-directions described above would be to efficiently generate a long key with the property that any subsequence of limited length, say l , would be provably secure, where one wants l to be as large as possible.

3 The Length of Keys

We note and foresee that increasing computer power is creating a growing gap between the key lengths of symmetric cryptosystems and of asymmetric cryptosystems. The effective key length of DES (1977, but still in use) is 56 bits. In the newly developed AES, we see a key length of 128 bits proposed, although also 192 and 256 bits are possible. Typically in symmetric systems, the effort to break the system grows exponentially in the key length. However, in most asymmetric systems the security grows sub-exponentially in the key length. This implies that faster computers in the future will lead to a much larger key expansion for asymmetric cryptosystems than for symmetric cryptosystems. This is highly unfortunate, since asymmetric systems have many natural advantages, most importantly the inherent possibility to distinguish the legitimate user from other parties involved, but also their use as one-way mappings. Any theoretical or practical technique to reduce this gap in size is of utmost importance.

Elliptic curve cryptosystems try to counter this problem. Their security also grows exponentially in the key length, but there are still many questions about their security. The XTR system creates an asymmetric system with a key length of say d , while the security is that of a corresponding discrete logarithms system with key length $3d$. Many approaches are possible and needed to reduce the complexity gap between symmetric and asymmetric cryptosystems, but a pessimistic point of view is that all future contributions will probably be very small.

A more challenging approach would be to find and describe other well-chosen subgroups of groups, such that the security parameters are related to the size of the group, while the complexity of the system is related to that of the subgroup. We note that investigations of the groups that have been proposed so far, e.g. class groups and matrix based systems, show great potential, but these investigations are far from complete. A systematic study of these systems, how they compare key-size-wise, security-wise, and performance-wise, would be valuable for the crypto community and could guide the direction of new and hopefully better alternatives. Applying the issues above to systems with added functionality, like being homomorphic

or having a homomorphic component, would open a completely new avenue of possibilities to explore.

In a parallel domain, namely that of secret key agreement schemes, we see a similar, but slightly different situation. Symmetric systems do not allow for secret key agreement. Asymmetric systems of course are very suitable for this purpose, but they rely on certain mathematical assumptions, like the difficulty of factoring. Secret key agreement systems that are provably secure are known in the literature. It seems reasonable to translate the concept of key in this setting to that of the amount of exchanged data. Asymmetric cryptosystems are much more efficient in this sense than the provably secure systems. Here we may see the opposite consequence of what the effect of faster computers will be! They will lead to a reduction of the gap between the complexity of asymmetric systems and unconditionally secure systems, for exactly the same reasons as explained before. This pleads for an increased study of unconditionally secure cryptosystems.