Topics and Research Directions for Symmetric Cryptography*

Alex Biryukov[1], Joan Daemen[2], Stefan Lucks[3], and Serge Vaudenay[4]

[1] SnT, CSC, University of Luxembourg,
[2] Radboud University, STMicroelectronics,
[3] Bauhaus-Universität Weimar,
[4] EPFL, Lausanne

This is a summary of the open discussion on future research topics for symmetric cryptography chaired by Stefan Lucks. During this session participants were suggesting topics of potential future interest. Here is the list of spotted topics. The first four were seen as somewhat more important, the rest are shown in the order they we called out.

Leakage Resilience: to offer protection against attacks that exploit leakage in implementations, one can design primitives, modes and protocols so that such leakage can either not be exploited or does not result in a compromise of security.

Beyond Birthday Security: the most widely used block ciphers have a block size of 128 bits (AES and most of its clones) or even 64 bits (Triple DES, IDEA and numerous so-called lightweight ciphers). As soon as the number of blocks processed with a block cipher with a given key approaches the square root of the domain size (2^{64} in the case of AES, 2^{32} in the case of DES), input collisions become likely and (distinguishing) security of breaks down for most modes. Some think that this should be addressed by having modes for which security would not break down at that point.

Lightweight Crypto: the participants at the discussion agreed that lightweight crypto should not be considered a single homogeneous field. There are at least two practical areas which demand lightweight cryptography, which have entirely different requirements:

- *Crypto for Low Power Devices:* battery-powered devices need crypto that requires little energy per cryptographic computation. Energy-harvesting devices need crypto that requires little power for cryptographic operations, while still achieving a certain throughput.

- *Low Latency Encryption:* this is required for implementing external memory encryption (useful against reverse engineering and in DRM) in a transparent way, i.e., without having to adapt the microprocessor architecture.

* The paper published in the proceedings of ESC'17 – Early Symmetric Crypto workshop, 2017.

- *Small Size (gate count) Crypto:* in the literature, the gate count (or the ”gate equivalence” from some hardware synthesis process) is often understood as a benchmark for new ”lightweight” primitives. It allows to compare different primitives (which one can be implemented with less gates?) and challenges both primitive designers and cryptanalysts (how many gates do you need to maintain a certain level of security?). However, during the discussion, no practical demand for extremely small symmetric primitives could be identified. Perhaps, the design and analysis of extremely small symmetric primitives is a nice intellectual challenge, but for practical purposes, more a distraction from the important issues for lightweight symmetric crypto, such as low power and low latency?

Standards/Correct Implementation: very often, successful attacks on cryptographic implementations do not arise from failures of cryptography by itself, but rather from seemingly silly implementation mistakes. E.g., the famous Heartbleed bug on OpenSSL was actually a simple buffer over-read error, from an incorrect implementation of the TLS protocol. To what degree can cryptographers contribute to proper specifications and correct implementation of the cryptography they propose?

Random Number Generators: building and evaluating random number generators does not strictly belong to the field of cryptography but is essential for many protocols.

Secure Channel Protocol: in symmetric cryptographic research there has been a shift from primitives (block ciphers, stream ciphers,) to more elaborate functions (authenticated encryption schemes). The next step up is secure channel protocols that satisfy more involved security requirements (e.g., when two parties share a symmetric key, how do they use is to secure a chat session between them?)

Post-Quantum Crypto: the impact of quantum computers on symmetric cryptography is not completely understood yet. Beyond the Grover algorithms, there exist ways to break symmetric schemes faster than with classical computers. The challenge is to develop more quantum attacks and identify countermeasures.

Format-Preserving Encryption: although NIST proposed a standard for encryption over small domains, the concept is not fully understood (namely, there is a big gap between the theory and practice when it comes to formalize security and prove it). In addition to this, current standards only encrypt over domains of form Z^ℓ, i.e. a string of ℓ elements of an alphabet Z. We should also construct ciphers over a more structured domains.

Low Multiplication Complexity Crypto: this is about symmetric crypto with minimal multiplicative size and depth. Such symmetric primitives could be useful for multi-party computation (MPC), homomorphic encryption (HE), and zero-knowledge proofs (ZK), where linear computations are very cheap compared to non-linear operations.

Block Ciphers for White-box Crypto: design a cipher for which it would be easy to produce white-box implementations (all white-box attempts for DES and AES have been broken). Another direction is to design provably weak white-boxed ciphers. This has been shown to be relatively easy, but doing it efficiently could be a challenge. Also design of strong white-box cipher (equivalent to public key) has been a long standing challenge.

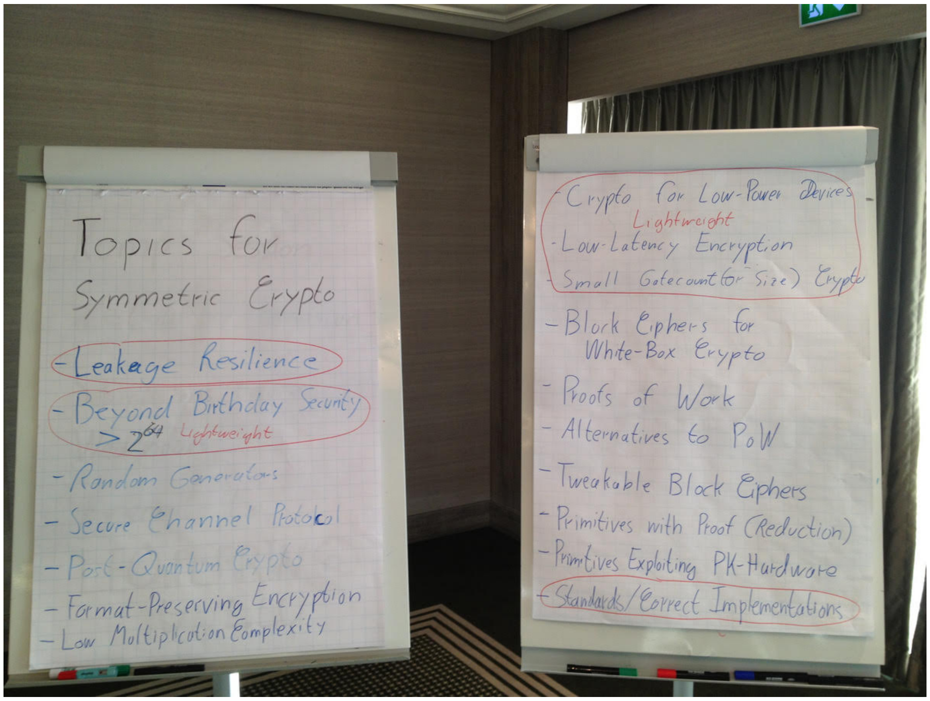
Proofs of Work (PoW): design functions which are provably hard (high computation and/or memory complexity) to evaluate. Such functions are useful in crypto-currencies, client puzzles (DDOS prevention) or as a mitigation against spam. Such functions are also closely related to hash functions used for password hashing.

Alternatives to PoW: since PoW functions are seen as wasting energy, green alternatives are always welcome. However these are typically protocols (proof-of-storage, proof-of-stake, byzantine fault tolerance) and wouldn't be in the realm of cryptography.

Tweakable Block Ciphers: over several decades, block ciphers have been the main workhorses of symmetric cryptography. Even cryptographic hash functions, such as MD4, MD5, SHA-1 and the SHA-2, family are based on an internal block cipher. One new primitive, challenging the ordinary block ciphers, are tweakable block ciphers (TBC). While it may be more challenging to design a secure TBC than a plain block cipher, and the TBC might be a little slower than the ordinary block cipher, at the same level of security, however the additional "tweak" input provides the mode and protocol designers with much greater degrees of freedom. This makes the primitive design and analysis of TBCs an important research question, just as the design and analysis of modes for TBCs.

Primitives with Proofs (Reductions): the "gold standard" for the design of asymmetric cryptographic primitives is to provide a security proof. The proof is a reduction, i.e., it shows that breaking the primitive is at least as hard as solving a certain algorithmic problem, which is meaningful beyond the cryptographic application. This is entirely different in symmetric crypto, where a primitive is claimed to be secure, and, at best, proven to withstand known attack techniques (differential cryptanalysis, linear cryptanalysis, ...). Is it possible to design symmetric primitives with a security proof, similarly to asymmetric cryptography? Of course, avoiding trivial PK to SK reductions or stream ciphers produced from one-way function hard bits, ideally keeping the traditional symmetric cryptography performance advantage – at least in part.

Primitives Exploiting PK-Hardware: on many platforms there are dedicated coprocessors to perform public key cryptography, such as a Montgomery multiplication unit. A block cipher (or permutation) using operations that can be performed efficiently on such a unit can give very good security at high speed.



Topics for Symmetric Crypto

- Leakage Resilience
- Beyond Birthday Security
  $> 2^{64}$ Lightweight
- Random Generators
- Secure Channel Protocol
- Post-Quantum Crypto
- Format-Preserving Encryption
- Low Multiplication Complexity

- Crypto for Low-Power Devices
  Lightweight
- Low-Latency Encryption
- Small Gatecount (or Size) Crypto
- Block Ciphers for White-Box Crypto
- Proofs of Work
- Alternatives to PoW
- Tweakable Block Ciphers
- Primitives with Proof (Reduction)
- Primitives Exploiting PK-Hardware
- Standards/Correct Implementations