Why IV Setup for Stream Ciphers is Difficult

Erik Zenner

Cryptico A/S
ez@cryptico.com

Dagstuhl, Jan. 8-12, 2007

Outline

Motivation

Attacks against IV setup are very successful against stream ciphers:

- The eStream project:
  - 34 stream ciphers submitted in May 2005
  - Thereof $\approx$ 20 broken by December 2006
  - Thereof $\approx$ 10 broken due to IV problems
- Other IV-based attacks:
  - WEP (RC4)
  - GSM (A5/1)
  - Turing, Helix,...

Engineering approach

In the following, we take an engineering approach.

- We are interested in constructing an object called "key/IV setup".
- In order to do so, we have to ask ourselves the following questions:
  - Security requirements and sound techniques for key/IV setup?
  - Security requirements and sound techniques for related objects?
  - Security requirements for a stream cipher?
  - What *is* a stream cipher?
- We deal with those questions in reverse order.

A disclaimer

- This talk is not about the outcome, but about the start of a research project.
  $\Rightarrow$ Many questions, few answers.
- If you know the answers, please let me know.
- If you don't know the answers, but think that the questions are interesting, then please let me know, too.
- If you don't know the answers and you think the questions are not interesting, then I apologize.

Outline

How it started...

Statement:

"Stream ciphers are always faster than block ciphers [...] but are considered much easier to break because of problems with the key handling and the pseudo nature of the number generators."

Chief Security Architect
of one of the world's largest companies
in summer 2005

- How to react to such a statement?
- My first intuition: You do not encrypt using block ciphers. You encrypt using block ciphers in a mode of operation. And those require initialization vectors, too.

Universal Secure Encryption Scheme

We have to make a proper comparison of encryption algorithms, based on what practitioners care about.

- Practitioners do not care for block ciphers or stream ciphers, for AES or RC4.
- Practitioners do not care for the difference between a cipher and a secure cipher.
- Practitioners care for a universal encryption algorithm that eats all kinds of input and encrypts it in a secure way.

USE Scheme:

In the following, I call this a **universal secure encryption (USE) scheme**. Note that it is not identical to the standard definition of a cipher or even a secure cipher.

Bellare and Rogaway's definition

Symmetric Encryption Scheme:

- A randomized **key generation algorithm**.
- A randomized or stateful **encryption algorithm** processing messages of arbitrary length.
- A deterministic **decryption algorithm**.

IND-CPA Security:

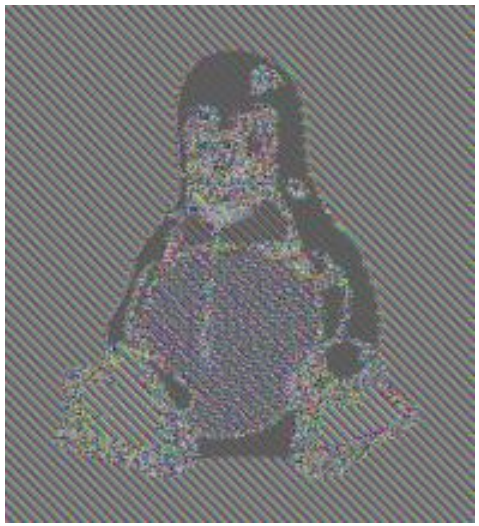
- Attacker chooses pairs of messages from $\{0,1\}^*$ and sends them to the encryption oracle.
- The oracle is either an L- or an R-oracle. An L-oracle encrypts the first, an R-oracle the second message.
- The attacker wins if he can tell whether he communicates with an L- or an R-oracle.

Some consequences

A corollary (Bellare, Rogaway):

"Any deterministic, stateless scheme is insecure."

- **All** USE schemes require initialization vectors (either as nonces or to remember the last state)!
- Block ciphers (also in ECB mode) are not USE schemes.
- How about stream ciphers?



What is a stream cipher? (1)

Narrow definition (informal):

A stream cipher consists of a pseudo-random bit generator (PRG) whose output is xored to the message in order to encrypt or decrypt.

Problem with this definition:

- Self-synchronizing stream ciphers?
- Stream ciphers with authentication?

$\Rightarrow$ Does not cover all stream cipher designs.

What is a stream cipher? (2)

Broad definition (Rueppel, HAC):

"**Stream ciphers** [...] encrypt individual characters [...] of a plaintext message one at a time, using an encryption transformation which varies with time. By contrast, **block ciphers** [...] tend to simultaneously encrypt groups of characters of a plaintext message using a fixed encryption transformation."

Problem with this definition:

- Distinguishing by block length is meaningless (modern stream ciphers have similar block lengths as stream ciphers).
- Statefulness turns out to be necessary condition for all secure schemes.

$\Rightarrow$ According to this definition, **all** USE schemes are stream ciphers!

Lessons

Lessons learned (1)

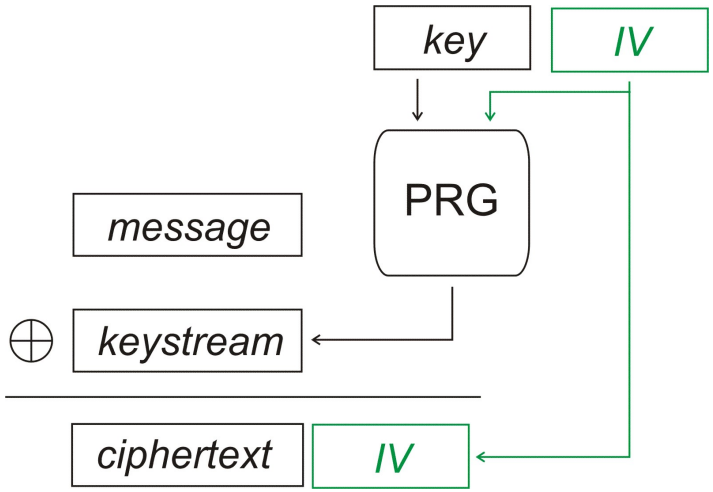
- A cipher is something different than a universal, secure encryption (USE) scheme.
- A block cipher is not a USE scheme and must not be used for encryption directly.
- The narrow definition of a stream cipher does not cover all designs.
- The broad definition of a stream cipher gives some minimum requirements that are necessary for **all** USE schemes and is thus useless.

Outline

Stream cipher definition

In the following:

- Use narrow definition of stream cipher.
- PRG expands key and IV into keystream.
- Keystream is xored to message to encrypt and to ciphertext to decrypt.

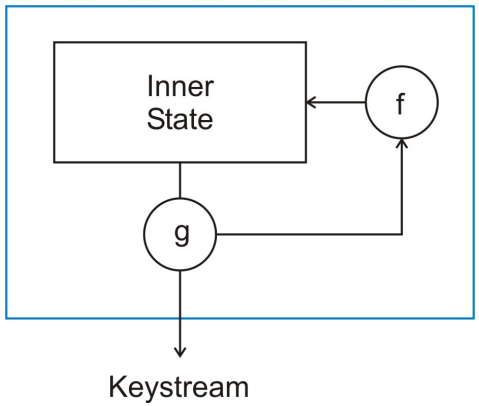


"Classical" PRG

"Classical" PRG:

- "key" is initial state.
- f is update function.
- g is output function.

Relatively well understood, but lacks key/IV setup.

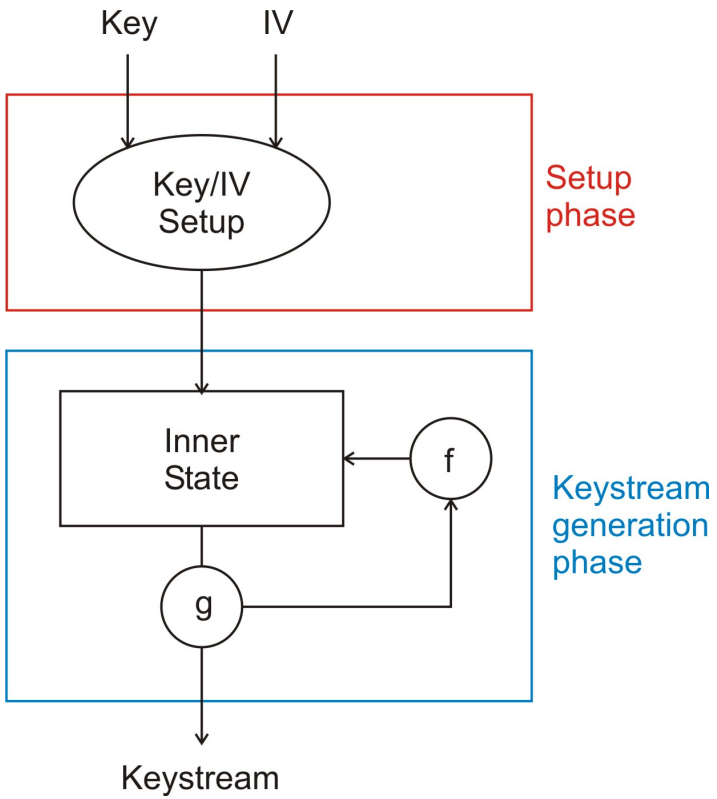


PRG-based stream cipher

PRG-based stream cipher:

- New component "key/IV setup".
- Has to mix key and IV into the inner state, but how?
- Security requirements? Construction principles?

Not well understood.



Stream cipher security

Starting point:

- Use the IND-CPA security definition for an encryption scheme.
- (Rogaway 2004) Give the adversary control over the IV, but make him nonce-respecting.

New IV-based attacks:

- (Hong, Sarkar 2005) An attack where several keys are attacked at once is a valid attack against all ciphers.
  - Disputed!
- (Wu, Preneel 2006) An attack where an IV is repeated can be a valid attack against certain stream ciphers.
  - Disputed!

In the following: Use security definition by Rogaway (2004), where a nonce-respecting adversary attacks only one key.

Outline

Modelling the key/IV setup

Security requirements (informal):

- Given the IV and a bitstream, the attacker must not be able to tell whether he sees a keystream or a random bitstream.
- If the PRG is cryptographically secure (and has a sufficient security level), then a sufficient condition is that the attacker can not distinguish the initial state from a random state.

In the following, we look for suitable building blocks, their definitions, construction principles, and efficiency.

First idea: Hash function

- **Intuition:** A hash function maps an input of arbitrary length onto an output of fixed length in such a way that the output "looks random".
- **Definition:** No known formalization describes the "looks random" property:
  - One-wayness, collision-resistance: Not sufficient in our case.
  - Random oracle: Suitable, but not universally accepted.
- **Construction principles:** Known problems with dedicated hash functions. Construction principles are less well-understood than previously believed.
- **Efficiency:** Might be less efficient than optimal:
  - Lack of key requires stronger construction.
  - Designed for long inputs.

Second idea: Key derivation function

- **Intuition:** A key derivation function (KDF) takes an input that has a secret and a public part and generates an output that can be used as a key (e.g. for a PRG).
- **Definition:** Long discussion on CFRG mailing list:
  - Most researchers have an intuition of what a KDF is, but a definite definition is lacking.
  - Possibly, the definition of a secure KDF with separate key and IV would be identical to that of a secure pseudo-random function (see next slide).
- **Construction principles:** There are only few designs for dedicated KDFs.
- **Efficiency:** ???

Third idea: Pseudo-random function

- **Intuition:** A pseudo-random function (PRF) maps a public input under a secret key onto an output that "looks random".
- **Definition:** PRFs have a well-understood security definition.
  - $\Rightarrow$ This means that we can solve the problem by modelling the key/IV setup as a PRF $K \times IV \rightarrow S$.
- **Construction principles:** There are almost no dedicated PRFs (mostly abused PRPs and derived constructions). Thus, they give limited help in constructing efficient practical solutions.
- **Efficiency:** PRFs might be more complicated than required here.
  - The majority of existing key/IV setup functions is too weak for a PRF.
  - Indistinguishability of initial state is a sufficient, but might not be a necessary criterion.

Outlook

Open question:

Can we find a more efficient cryptographic building block that would also be sufficient?

- In order to do that, we have to take the properties of the PRG into account.
- Good formalizations of PRGs exist only in the asymptotical model...

Lessons

Lessons learned (2)

- We do not have a security definition for what an efficient, secure IV setup has to do.
- For many candidate building blocks, we face at least one of the following problems:
  - Security definitions unclear.
  - No dedicated constructions exist.
  - Existing dedicated constructions insecure.
- A pseudo-random function will solve the problem, but might be computational overkill.
- In order to do better, we have to take the PRG into account, which again is not well-understood.

Outline

Conclusions

Why IV setup for stream ciphers is difficult:

- We do not know what we mean by a stream cipher.
- Even for a narrow definition of a stream cipher, we do not agree on what constitutes a valid attack.
- Even for a narrow definition of an attack, we do not know what the key/IV setup is supposed to achieve.
- Even for related building blocks, we do not really agree on what they are supposed to achieve, and how to construct them securely and efficiently.