GMFH, 19.6. 2009

Cryptography: An Overview

Willi Meier

n|w Fachhochschule Nordwestschweiz
Hochschule für Technik

Overview

- Where is cryptography implemented?
- Topics of this workshop
- Projects at FHNW
- Hash functions
- The NIST SHA-3 hash function competition
- Hash function BLAKE
- Stream Ciphers
- The eSTREAM Project
- Two eSTREAM finalists: Grain and Trivium
- Cryptanalysis Scenario

Cryptography implemented in:

- Secure computer networks (including Internet)
- Electronic fund transfer, eBanking
- Secure mobile phones
- Embedded systems
- Biometric passports
- Secure RFID's
- Anonymity of electronic data

Topics to be presented

Anonymization

A system developed by David-Olivier Jaquet-Chiffelle allows the federal statistical office to conduct an exhaustive statistics of hospitalized people (with in particular recognition of multiple hospitalizations) while guaranteeing the anonymity of the patients.

The solution is based on crytographic primitives.

This is an interesting example on how to use cryptography in order to allow a secure treatment of sensitive statistical data.

Algebraic methods for cryptanalysis

In cryptology, the field of cryptanalysis is concerned with methods for recovering the secret key of some cipher.

One class of methods views the cipher as an ensemble of algebraic equations over some finite field, typically GF(2) or an extension of it.

In his talk, Jean-Philippe Aumasson will present a recent type of algebraic attack, called cube attack, which is a general purpose method for analyzing various types of cryptographic algorithms.

He will also describe some concrete applications of this attack to real-world ciphers.

Elliptic Curve Cryptography

Elliptic curves are a well established topic in algebraic number theory.

They have important applications in public key cryptography:

In public key cryptography, there exist systems whose security is related to the discrete logarithm problem over the integers modulo a large prime. This prime modulus has to be large in order that discrete logs cannot be solved.

It is believed that the discrete log problem is much harder when modular arithmetic is replaced by arithmetic in an elliptic curve over a finite field.

This allows for efficient operations over smaller finite fields.

Due to this advantage, elliptic curves are implemented in many environments.

BouncyCastle is an open source Crypto provider written in Java which supplies classes for Elliptic Curve Cryptography (ECC).

Daniel Mall has found a flaw in the class ECPoint resulting from an unhappy interaction of elementary algorithms.

He shows how to exploit this flaw to a real world attack, e.g., on the encryption scheme ECIES.

FHNW projects related to Crypto

Design of new, and analysis of existing systems

Contributions to International Projects:

NIST SHA-3

ECRYPT NoE eSTREAM

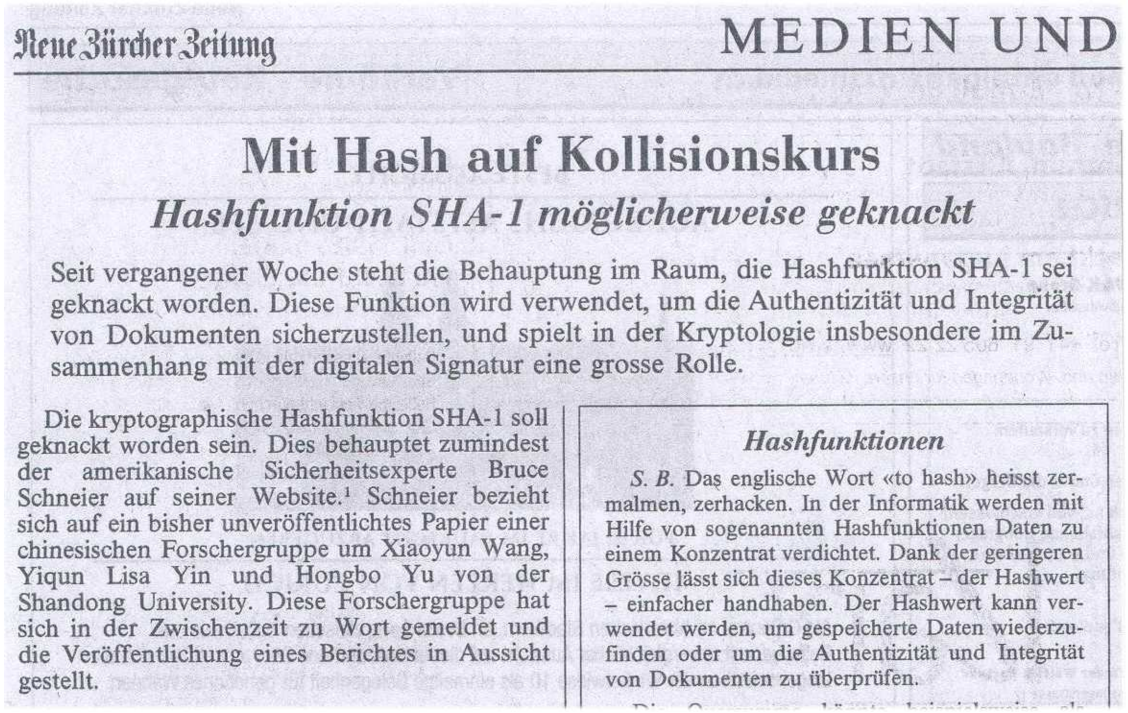
Funded by: FHNW, SNF, GEBERT RÜF, Hasler, CTI

Some practically implemented crypto systems not secure anymore:

Bluetooth Protocol

Digital Signatures, use insecure hash functions

Hash Functions



Neue Zürcher Zeitung

MEDIEN UND

Mit Hash auf Kollisionskurs

Hashfunktion SHA-1 möglicherweise geknackt

Seit vergangener Woche steht die Behauptung im Raum, die Hashfunktion SHA-1 sei geknackt worden. Diese Funktion wird verwendet, um die Authentizität und Integrität von Dokumenten sicherzustellen, und spielt in der Kryptologie insbesondere im Zusammenhang mit der digitalen Signatur eine grosse Rolle.

Die kryptographische Hashfunktion SHA-1 soll geknackt worden sein. Dies behauptet zumindest der amerikanische Sicherheitsexperte Bruce Schneier auf seiner Website.[1] Schneier bezieht sich auf ein bisher unveröffentlichtes Papier einer chinesischen Forschergruppe um Xiaoyun Wang, Yiqun Lisa Yin und Hongbo Yu von der Shandong University. Diese Forschergruppe hat sich in der Zwischenzeit zu Wort gemeldet und die Veröffentlichung eines Berichtes in Aussicht gestellt.

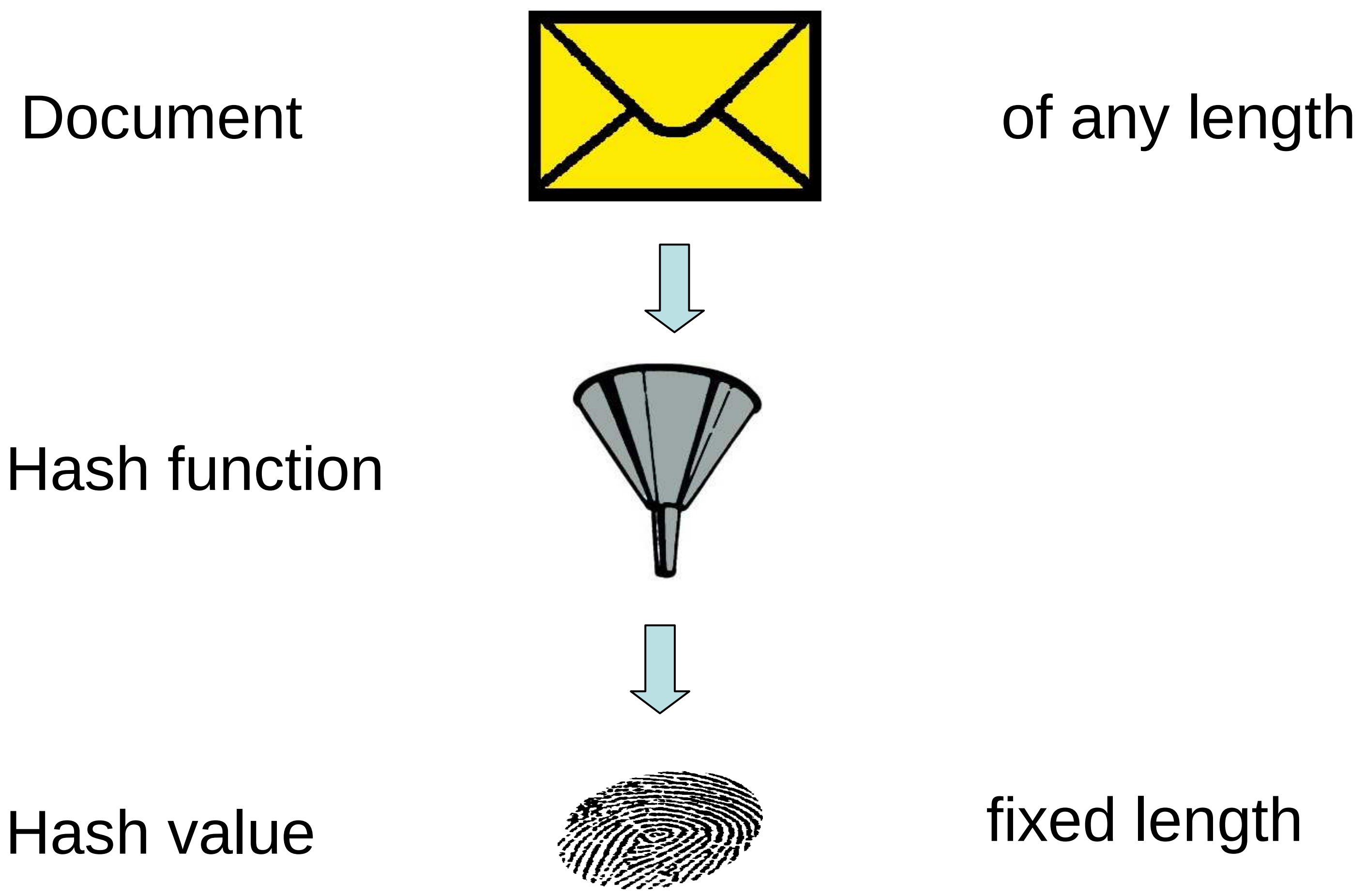
Hashfunktionen

S. B. Das englische Wort «to hash» heisst zermalmen, zerhacken. In der Informatik werden mit Hilfe von sogenannten Hashfunktionen Daten zu einem Konzentrat verdichtet. Dank der geringeren Grösse lässt sich dieses Konzentrat – der Hashwert – einfacher handhaben. Der Hashwert kann verwendet werden, um gespeicherte Daten wiederzufinden oder um die Authentizität und Integrität von Dokumenten zu überprüfen.

Where are hash functions used?

1. Digital Signatures
2. Integrity check of data
3. Secure passwords
4. Pseudo random generators
5. …..

Principle


Document
of any length
Hash function
Hash value
fixed length

Hash functions map a digital message of any length into a bit string of fixed length:

$$H\text{: message} \rightarrow \text{hash value}$$

Typical hash functions have a hash value of length 160 bits, 256 bits or 512 bits.

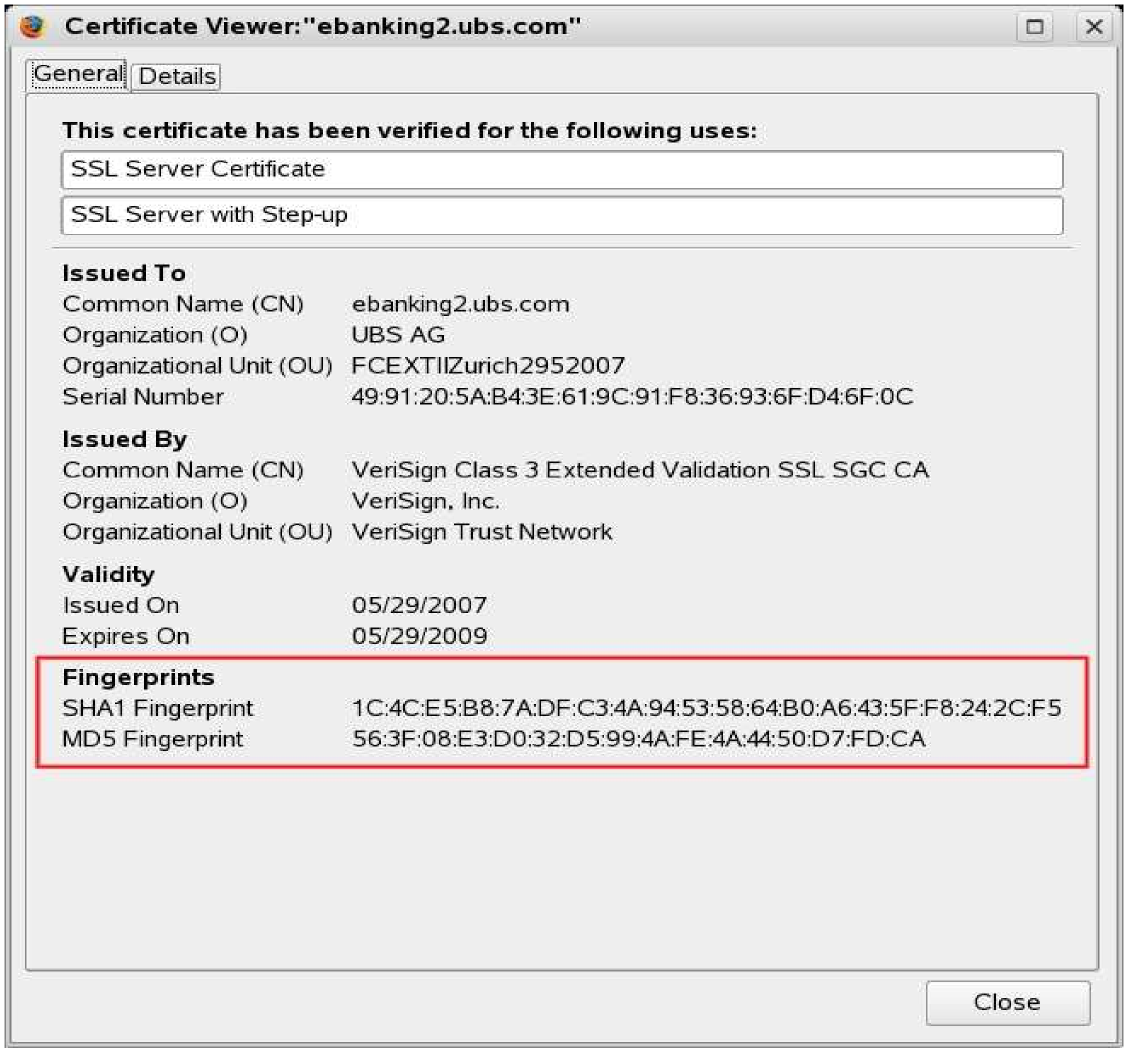
Important: A small change in a message should have a large effect on its hash value.

Example: MD5

H("Hello") = 09f7e02f1290be211da707a266f153b3

H("Hallo") = 3290ec3c19a8a39362f7d70043f15627

Example: UBS Certificate



Certificate Viewer:"ebanking2.ubs.com"

General | Details

This certificate has been verified for the following uses:

SSL Server Certificate

SSL Server with Step-up

Issued To

Common Name (CN)	ebanking2.ubs.com
Organization (O)	UBS AG
Organizational Unit (OU)	FCEXTIIZurich2952007
Serial Number	49:91:20:5A:B4:3E:61:9C:91:F8:36:93:6F:D4:6F:0C

Issued By

Common Name (CN)	VeriSign Class 3 Extended Validation SSL SGC CA
Organization (O)	VeriSign, Inc.
Organizational Unit (OU)	VeriSign Trust Network

Validity

Issued On	05/29/2007
Expires On	05/29/2009

Fingerprints

SHA1 Fingerprint	1C:4C:E5:B8:7A:DF:C3:4A:94:53:58:64:B0:A6:43:5F:F8:24:2C:F5
MD5 Fingerprint	56:3F:08:E3:D0:32:D5:99:4A:FE:4A:44:50:D7:FD:CA

Close

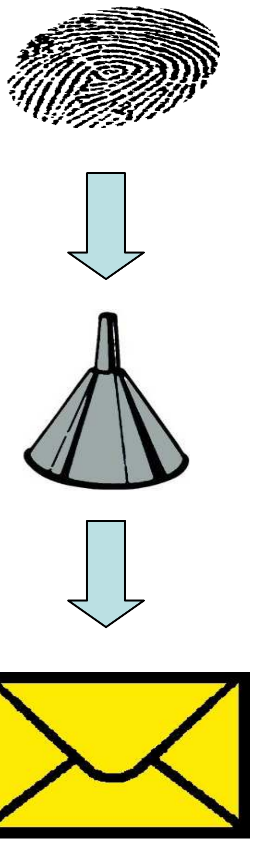
Security requirements

It should be infeasible to find a **preimage**:

hash value

inversion

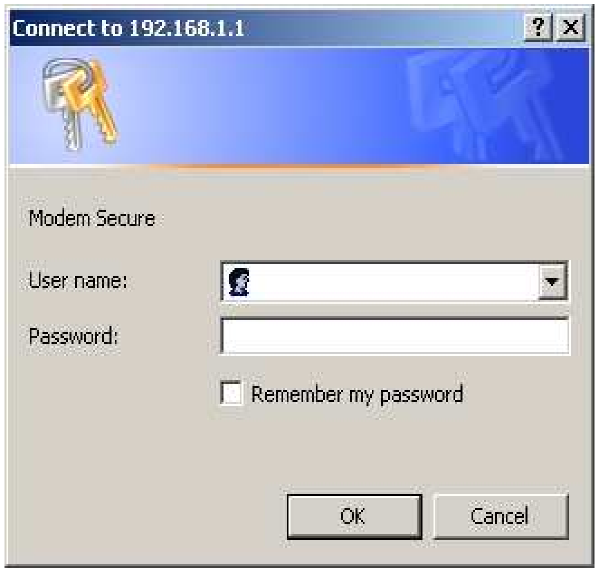
document



many preimages

Security of passwords

Login with User ID and Password


Connect to 192.168.1.1

Modem Secure

User name:

Password:

Remember my password

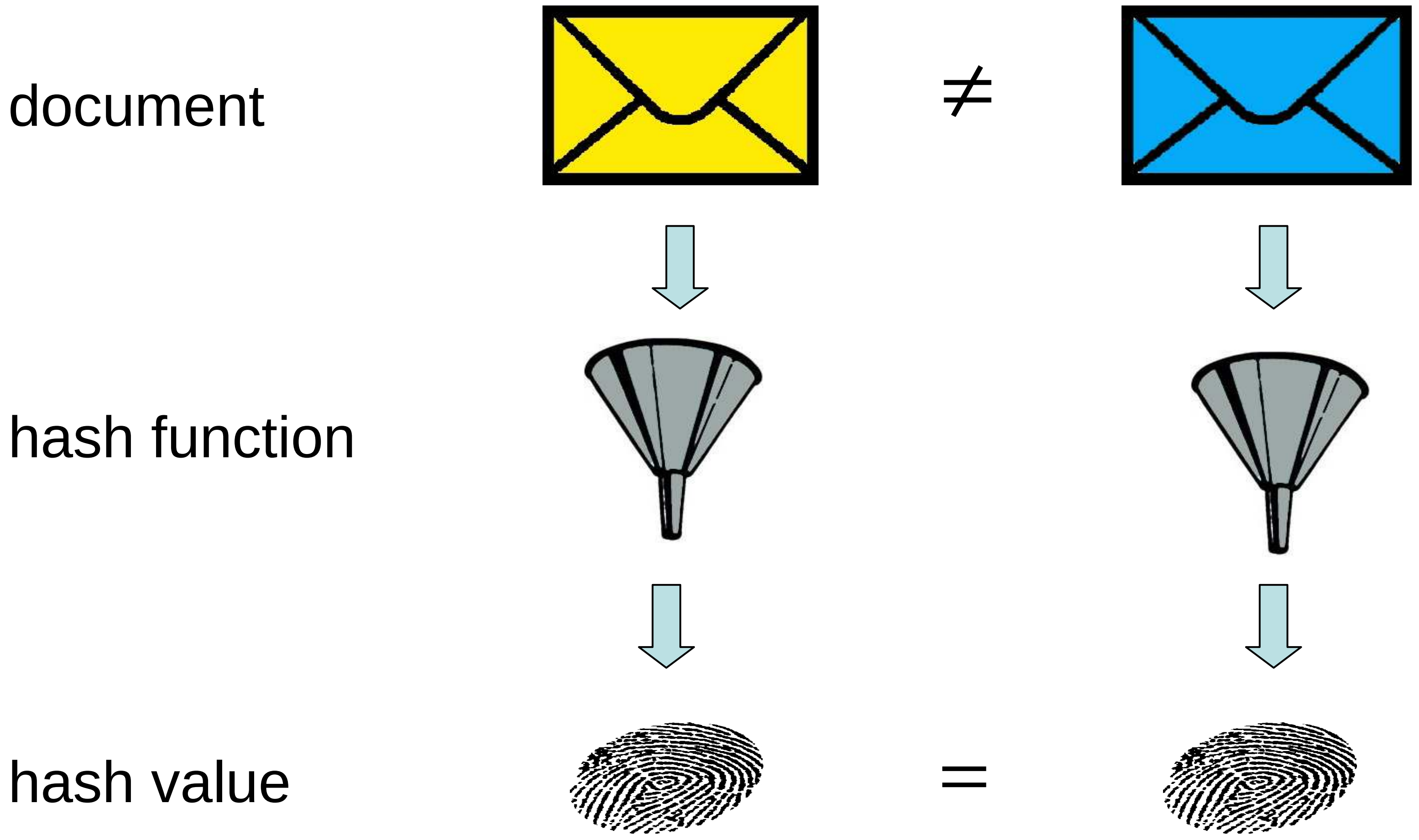
Computer stores fingerprint of the password, and checks, whether the entered password has the correct fingerprint.

Illegitimate user has no access to computer.

If preimages can be found, the password is insecure.

Security requirement

It should be infeasible to find a **collision**:



Digital Signature

Handwritten Signature

[illegible]

Digital Signature

011101….00110100111

In **digital signatures,** document is not signed directly, but its fingerprint (for efficiency reasons).

If for a given document another document with the same fingerprint exists, forging of signatures becomes possible:

Example of a critical collision:

H("I transfer to XY the amount of 100 CHF") = H("I transfer to XY the amount of 100000 CHF")

What is the meaning of collision resistance?

model:

- Inputs of a hash function: balls
- Outputs of a hash function: set of bins
- Hash function throws balls into bins

If there are more balls than bins, at least one bin gets more than one ball.

Birthday paradox

N bins

If there are more than sqrt(N) balls:

With probability greater than ½ two balls are thrown into a same bin.

A hash function with n binary outputs is called **collision resistant**, if the complexity to find a collision is not significantly smaller than $2^{n/2}$ computations of the hash function.

A hash function is **broken** if there is an algorithm that can find collisions with much smaller effort.

Status of hash functions

NIST Standards and MD5 (are in worldwide use):

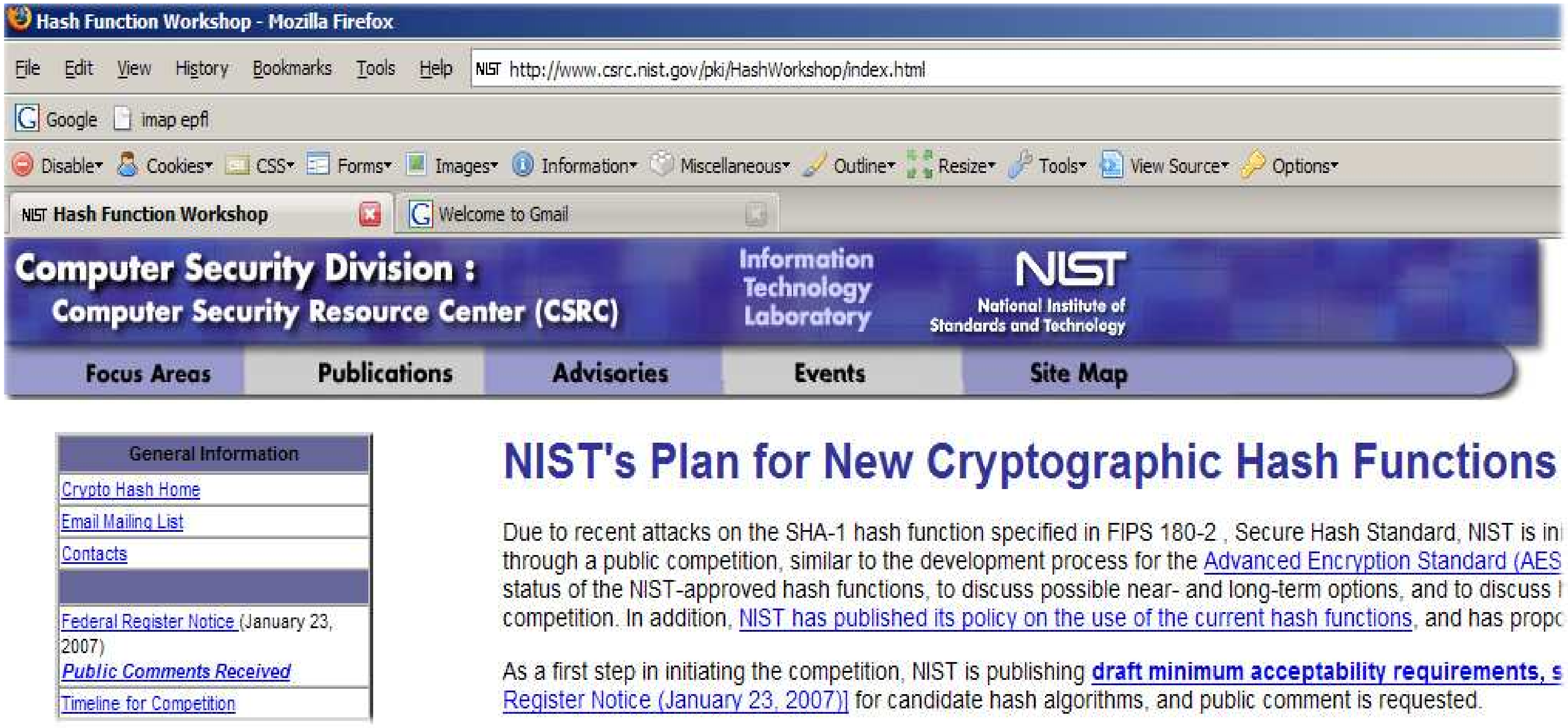
MD5, SHA-1 broken (Wang 2005).

SHA-2 unbroken,

but all these hash functions rest on similar components for their design.

Unclear security principles.

Reaction: NIST SHA-3 project


Hash Function Workshop - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://www.csrc.nist.gov/pki/HashWorkshop/index.html
Google imap epfl
Disable Cookies CSS Forms Images Information Miscellaneous Outline Resize Tools View Source Options
Hash Function Workshop
Welcome to Gmail
Computer Security Division :
Computer Security Resource Center (CSRC)
Information Technology Laboratory
NIST National Institute of Standards and Technology
Focus Areas
Publications
Advisories
Events
Site Map
General Information
Crypto Hash Home
Email Mailing List
Contacts
Federal Register Notice (January 23, 2007)
Public Comments Received
Timeline for Competition

NIST's Plan for New Cryptographic Hash Functions

Due to recent attacks on the SHA-1 hash function specified in FIPS 180-2 , Secure Hash Standard, NIST is ini
through a public competition, similar to the development process for the Advanced Encryption Standard (AES
status of the NIST-approved hash functions, to discuss possible near- and long-term options, and to discuss l
competition. In addition, NIST has published its policy on the use of the current hash functions, and has propo

As a first step in initiating the competition, NIST is publishing **draft minimum acceptability requirements, s**
Register Notice (January 23, 2007)] for candidate hash algorithms, and public comment is requested.

International competition NIST SHA-3 for new hash functions (2008-2012).

64 proposals were submitted.

51 first round submissions accepted by NIST.

About a dozen of these broken so far.

NIST plans to reduce the 51 down to about 15 candidates for 2nd round.

Hash function BLAKE

Own design of a SHA-3 candidate hash function, jointly with Jean-Philippe Aumasson, Luca Henzen (ETHZ) and Raphael Phan, Loughborough University.

Software-oriented.

Based on design architecture different from MD5 or SHA.

Uses only interaction of three basic operations:

integer addition modulo 2^w, w: word length

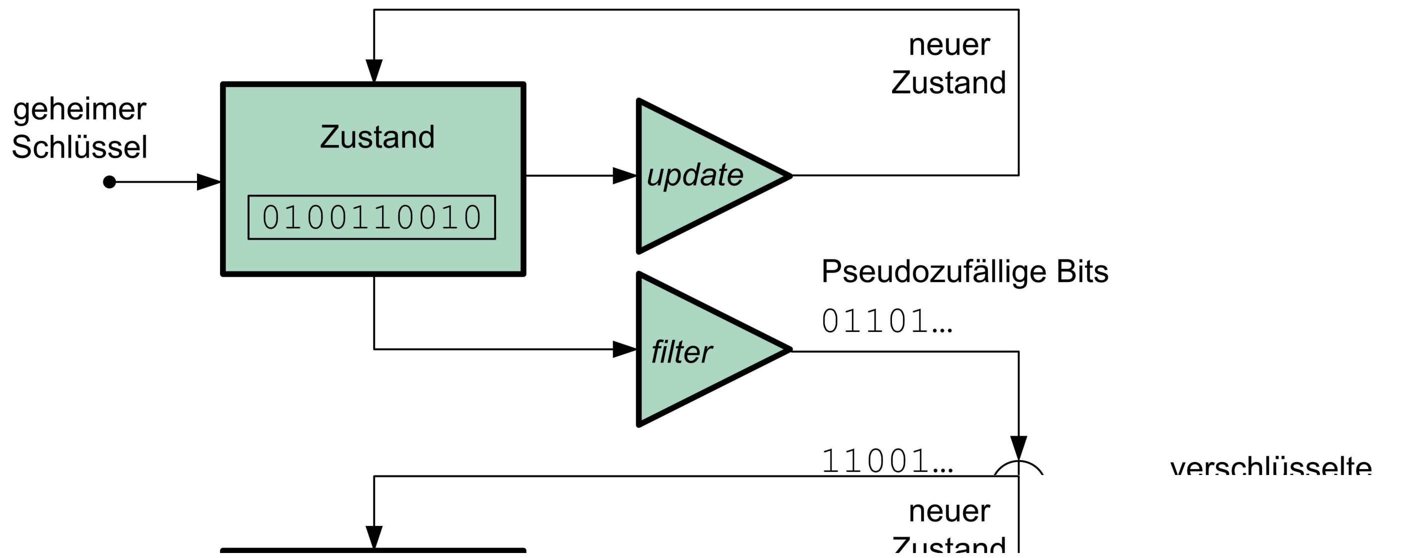
addition modulo 2

and circular rotations of bits in a word, $w = 32$ or 64.

Own security analysis of competitor designs.

What is a Stream Cipher?

Aim is confidentiality of messages. Stream Ciphers are small and fast.



Profile 1: Optimized for software (high throughput).
Profile 2: Optimized for small hardware.

Implemented in:

Mobile phones

RFID's

Known stream ciphers:

RC4, in Netscape Secure Socket Layer (SSL) Protocol, but also in eBanking.

A5, in Global System for Mobile Communication (GSM).

Bluetooth Stream Cipher, Standard for wireless short-range connectivity.

Prototype of a stream cipher: One-Time-Pad

Keystream purely random bit string of same length as message.

May be used only once, otherwise insecure.

If used properly, One-Time-Pad unconditionally secure.

Disadvantage: Secure distribution of a long secret key.

In practical applications: Replace purely random keystream by output stream of a pseudorandom generator.

Initial state: Short purely random bit string K (e.g., of length $n = 128$ bits).

Instead of long keystream, only this short bit string needs to be securely transmitted.

Provable security is lost.

Effort to determine K from known output string should be close to 2^n computations.

Most pseudorandom generators for computer simulations won't satisfy such a high requirement.

"Practical Security": Thus far, no one hash found an attack.

Many stream ciphers broken in the past.

The eSTREAM Project

eSTREAM was a project with the aim: "to identify new stream ciphers that might become suitable for widespread adoption".

Organized by EU ECRYPT network of excellence.

Call for Primitives 2004

Project finalized 2008

Various project phases

Two profiles for eSTREAM:

Profile 1: Stream ciphers for software applications with high throughput (faster than block cipher AES in counter mode).

Profile 2: Stream ciphers for hardware implementations in constrained resources, e.g., restricted memory, small hardware, restricted power consumption.

Reaction to call: submission of 34 proposals!

As of 2009:

Four finalists for software profile

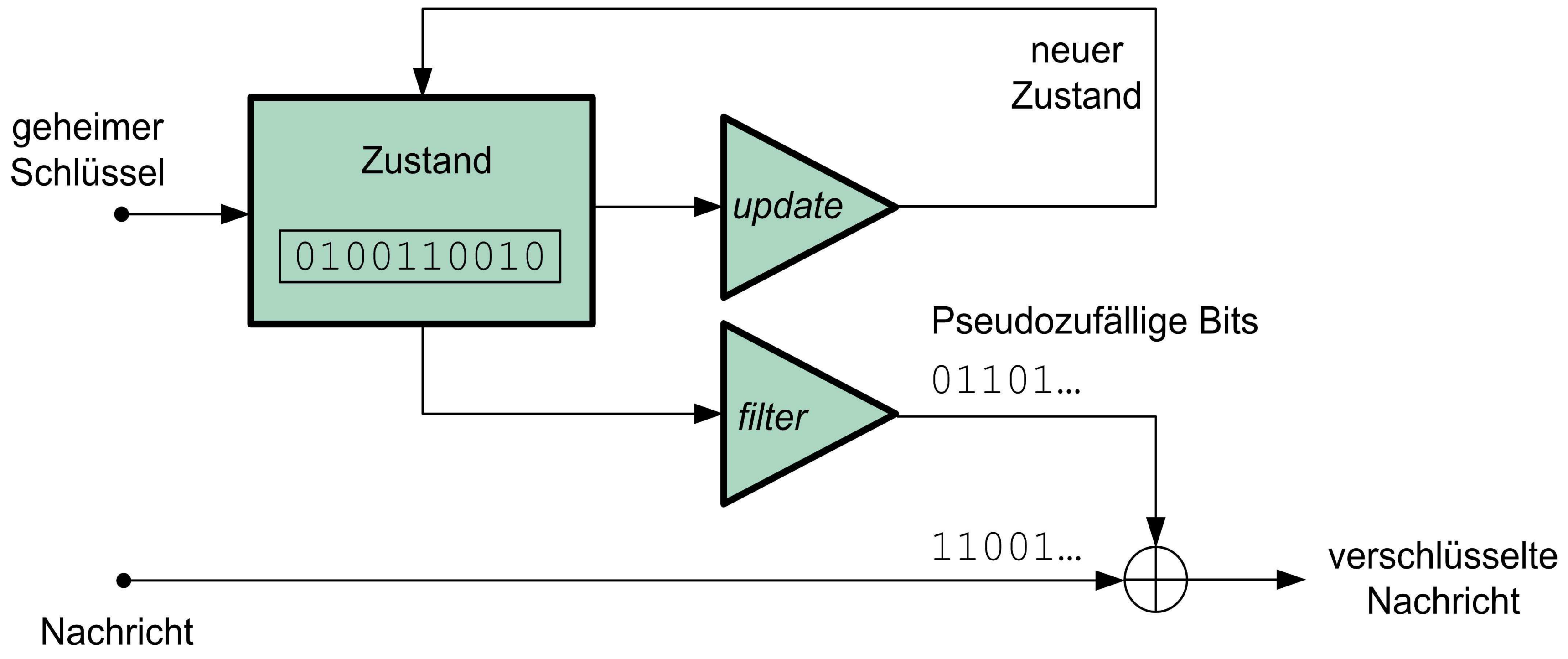
Four finalists for hardware profile. One of these severely broken meanwhile.

Construction principles for dedicated stream ciphers

State with update function

Output- or filter function, defined on present state

Update of state linear or nonlinear


neuer Zustand
geheimer Schlüssel
Zustand
0100110010
update
Pseudozufällige Bits
01101...
filter
11001...
verschlüsselte Nachricht
Nachricht

State and update in hardware-oriented stream ciphers often one or more linear feedback shift registers (LFSR).

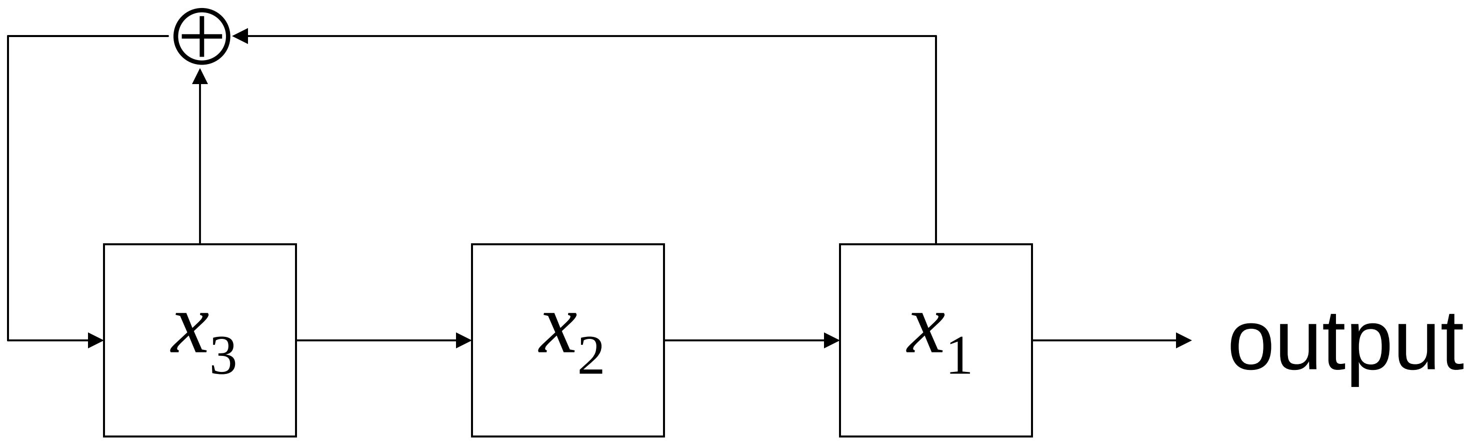
LFSR of length L:
Consists of a bit vector $(x_1,\ldots,x_L)$. In each step, every state bit is shifted by one position to the right, except the rightmost bit x_1, which is output.

On the left, a new bit is shifted in using a linear recursion

$$x_j = (c_1 x_{j-1} + c_2 x_{j-2} + \ldots + c_L x_{j-L}) \bmod 2,$$

for j greater than L.

Example: $L = 3$



$$x_j = x_{j-1} + x_{j-3} \bmod 2$$

Depending on the chosen linear recursion, LFSR‘s have desirable properties:

- Produce output sequences of large period (e.g. maximum period 2^n-1)
- Produce sequences with good statistical properties
- Can be readily analyzed using mathematics in finite fields.
- Disadvantage: For known output sequence, initial state can be easily found by solving system of linear equations.

Linear recursion of LFSR can be described by feedback polynomial:

For an n-stage LFSR with feedback coefficients $c_0, c_1, \ldots, c_n$, the characteristic polynomial is defined by

$$f(x) = c_0 + c_1 x + \ldots + c_{n-1} x^{n-1} + x^n$$

f is primitive, if f divides $x^{2^n-1} + 1$, but not

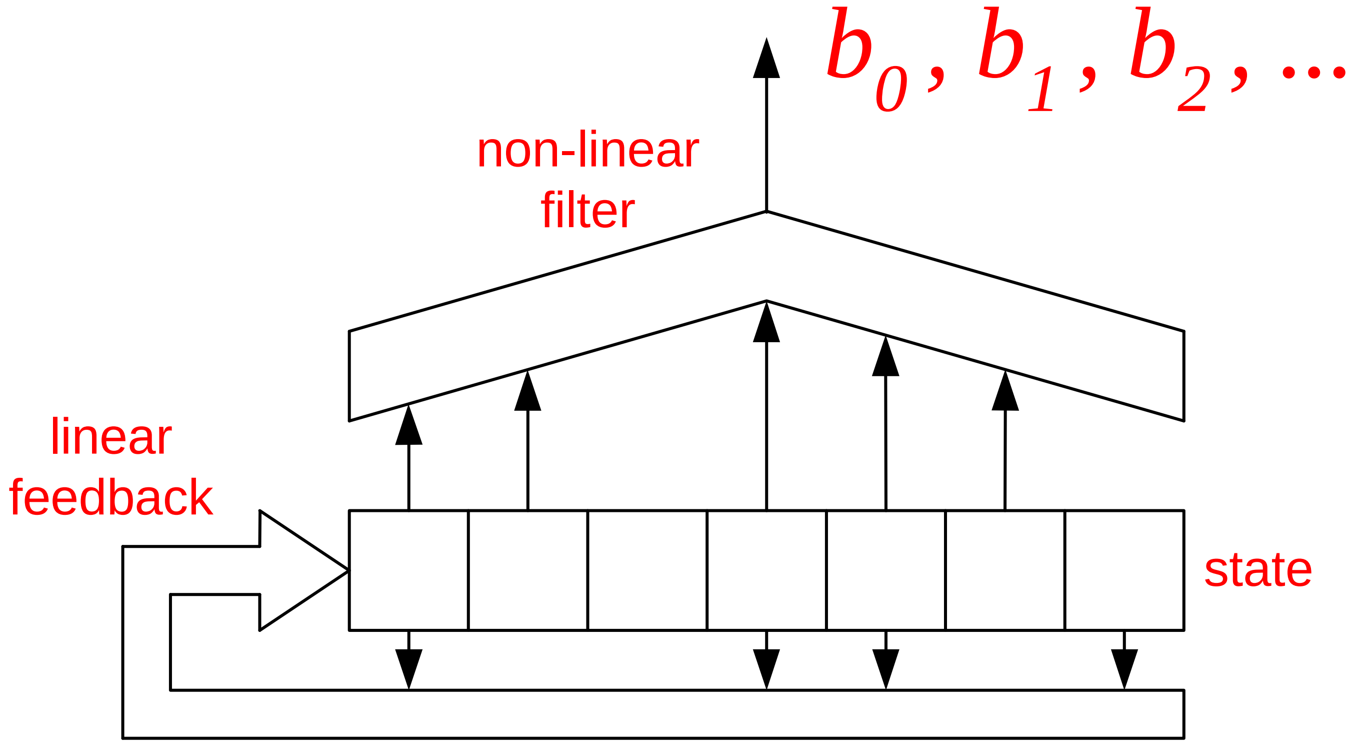
$x^e + 1$ with $e < 2^n - 1$. (Polynomial arithmetic

over GF(2)).

If f is primitive, output sequence of LFSR has maximum period $2^n - 1$.

Method to destroy nonlinearity: Nonlinear filter generator

Generates keystream bits b_0, b_1, b_2 ,..., using nonlinear function f defined on state bits of LFSR.



Example of a generator with 3 LFSR's combined: Geffe Generator (remains only of historical interest).

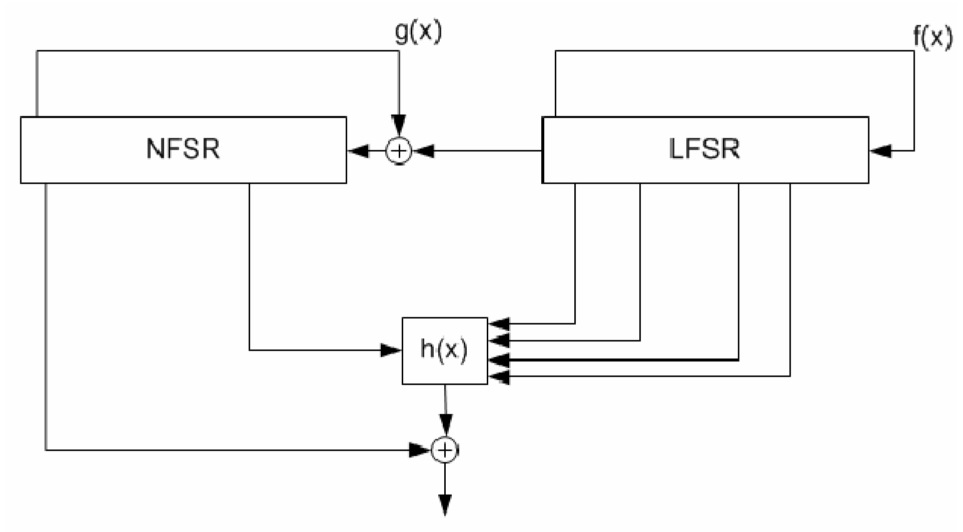
3 LFSR‘s X, Y, Z, with outputs x_t, y_t, z_t.

Output b_t of generator determined by:

If $y_t = 1$
$b_t = x_t$
else
$b_t = z_t$

Combining function:
$b = f(x,y,z) = x_t * y_t+(y_t+1) * z_t$.

Own Design (with Uni Lund): Grain



eSTREAM finalist. Has 3 components:
80 bit LFSR, 80 bit nonlinear feedback shift register NFSR, nonlinear filter h.

Input in NFSR masked by a LFSR bit.

Output bit masked by xor of 7 NFSR bits.

Grain-128

Variant with 128-Bit key and two 128 bit registers.

Grain allows for increased efficiency due to parallelization of hardware.

Grain, as most dedicated stream ciphers, works with a (public) initial value IV

Allows to reuse the same key for multiple transmission of secure data, and for resynchronization purposes.

Trivium

State: 288 Bits

Nonlinear update

Linear output function

80-bit key

State consists of 3 registers,
$R_1 = (x_1,\ldots,x_{93}),\ \ R_2 = (x_{94},\ldots,x_{177}),\ \ R_3 = (x_{178},\ldots, x_{288}).$

Update and Output of Trivium

$$t_1 \leftarrow x_{66} + x_{93}$$

$$t_2 \leftarrow x_{162} + x_{177}$$

$$t_3 \leftarrow x_{243} + x_{288}$$

$$z_i \leftarrow t_1 + t_2 + t_3$$

$$t_1 \leftarrow t_1 + x_{91}x_{92} + x_{171}$$

$$t_2 \leftarrow t_2 + x_{175}x_{176} + x_{264}$$

$$t_3 \leftarrow t_3 + x_{286}x_{287} + x_{69}$$

$$(x_1, ..., x_{93}) \leftarrow (t_3, x_1, ..., x_{92})$$

$$(x_{94}, ..., x_{177}) \leftarrow (t_1, x_{94}, ..,..., x_{176})$$

$$(x_{178}, ..., x_{288}) \leftarrow (t_2, x_{178}, ..., x_{287})$$

Cryptanalysis Scenario:

Assume that opponent knows parts of message (plaintext), or he knows that message is encoded in ASCII and thus has redundancy .

In stream cipher: Known plaintext is equivalent to known output segment, i.e., known keystream: ciphertext is obtained out of plaintext by bitwise xor with keystream.

In cryptanalysis of stream ciphers it is therefore assumed that part of keystream is known.

More to cryptanalysis of stream ciphers: Talk by Jean-Philippe Aumasson.