Edon-*R*, An Infinite Family of Cryptographic Hash Functions

Danilo Gligoroski[1], Smile Markovski[2], Ljupco Kocarev[3]

1. Centre for Quantifiable Quality of Service in Communication Systems, Q2S - NTNU, NORWAY
2. Faculty of Natural Sciences, Univ. Ss Cyril and Methodius – Skopje, MACEDONIA
3. University of California San Diego, Institute for Nonlinear Science, USA

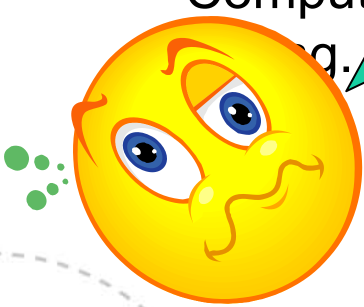
Outline

- **Design principles for Edon-*R***
- **One-way functions based on quasigroup transformations**
- **Definition of Edon-*R***
- **Some properties of Edon-*R***
- **Conclusions**

Design principles for Edon-*R*

1. **Huge family instead of one function**
   – *"If a significant number of messages can be attacked simultaneously, it is advisable to select a larger value of n. In that case it is preferable to use a UOWHF rather than a simple OWHF: as every instance will have a different parameter, this prevents an attacker from amortizing his effort over many targets."* – B. Preneel, *The State of Cryptographic Hash Functions,* Lectures on Data Security, Lecture Notes in Computer Science 1561, pp. 158-182, 1999, Springer-Verlag.

Why, still, almost every new practical design of a cryptographic hash function is simple OWHF and not UOWHF??!?



Design principles for Edon-*R* (cont.)

2. A design that produces hash outputs of various length instead of a fixed one

Because:

- Ever increasing computing power makes obsolete some designs of hash functions.
- We want to avoid having to design and redesign cryptographic hash functions with bigger and bigger hash output every 15 years.
- The use of hash functions that produce 256 or 512 bits is not always optimal from the speed point of view.

Design principles for Edon-*R* (cont.)

2. A design that produces hash outputs of various length instead of fixed one (cont.)

Compare the development of two cryptographic primitives during the last 15-20 years: RSA and MDx family of hash functions:

- RSA – one design, capable to operate with different lengths of the prime numbers. As computing power has increased, the length of prime numbers has increased, but the basic algorithm remains the same.
- MD4 was almost immediately replaced by MD5, then SHA-0 almost immediately replaced by SHA-1, and SHA-2 is now introduced as a replacement function.

Design principles for Edon-*R* (cont.)

3. **A design where the compression function is one-way candidate function based upon some hard mathematical problem instead of ad-hoc complex operations**
   - Edon-*R* is based on theory of quasigroups and quasigroup string transformations and quasigroup one-way candidate functions.
   - Its cryptographic strength relies upon the hardness of solving non-linear systems of quasigroup equations.
   - Quasigroups in general are algebraic structures with one binary operation which do not satisfy the usual algebraic laws used in solving equations (the commutative law, the associative law, the idempotent law, having zeros or units, and so on).

Design principles for Edon-*R* (cont.)

3. A design where the compression function is one-way candidate function based upon some hard mathematical problem instead of ad-hoc complex operations (cont.)

   - Similar approaches:
     - Damgaard – 1988 (intractability of discrete logarithm problem)
     - Gibson – 1991 (intractability of discrete logarithm problem)
     - Contini, Lenstra and Steinfeld – 2005 (hardness of the number factorization problem)

   - Common disadvantage: SLOW COMPUTATIONAL SPEED

Design principles for Edon-*R* (cont.)

4. **The size of the internal memory of the iterated compression function of Edon-*R* is twice the size of its hash output.**
   - Latest breakthroughs in theoretical understanding of iterated hash functions (multi-collisions)
     - Joux (2004)
     - Kelsy and Schneier (2005)
   - Design suggestions by
     - Lucks (2004)
     - Coron, Dodis, Malinaud and Puniya (2005)

Summary of design principles of Edon-*R*

1. Huge family (UOWHF) instead of one OWHF.
2. A design that produces hash outputs of various length instead of a fixed one.
3. A design where the compression function is one-way candidate function based upon some hard mathematical problem instead of ad-hoc complex operations.
4. The size of the internal memory of the iterated compression function of Edon-*R* is twice the size of its hash output.

One-way functions based on quasigroup transformations

Q2S

Brief description:

(Q,*)

*	0	1	2	3
0	3	1	0	2
1	0	3	2	1
2	2	0	1	3
3	1	2	3	0

The *R*1 is one-way candidate function.

Reverse order

R_1	a_1	a_2	a_3	a_4
a_4	b_{11}	b_{12}	b_{13}	b_{14}
a_3	b_{21}	b_{22}	b_{23}	b_{24}
a_2	b_{31}	b_{32}	b_{33}	b_{34}
a_1	b_{41}	b_{42}	b_{43}	b_{44}

If (Q,*) is shapeless, then the final string $b_{41}b_{42}b_{43}b_{44}$ is easy to compute in one direction (from a given string $a_1a_2a_3a_4$), but it is hard to find its preimage.

(Q,*) is a **Shapeless quasigroup** of order n if it is non-commutative, non-associative, it does not have neither left nor right unit, it does not contain proper sub-quasigroups and there is no $k < 2n$ for which the identities of the kinds:

$$\underbrace{x*(x*\cdots*(x}_{k}*y)\cdots)=y \qquad y=(\cdots\underbrace{(y*x)*\cdots*x)}_{k}$$

are satisfied.

NTNU
Innovation and Creativity

One-way functions based on quasigroup transformations

Theorem 1.

If the quasigroup (Q,*) *of order* n *is shapeless, then the number of computations based only on the lookup table that defines the quasigroup* (Q,*) *for finding a preimage of the function* $R_1 : Q^r \rightarrow Q^r$ *is* $O(n^{\lceil r/3 \rceil})$.

One-way functions based on quasigroup transformations

- **Hypothesis** *“There is no effective algorithm for solving a system of non-linear quasigroup equations in a shapeless quasigroup.”*
- The quasigroup string transformations can be seen as a special type of cellular automata operations. The (un)predictability of cellular automata was investigated by Moore et al. in 1997 and 2000 in cases when the obtained quasigroups have richer structure than that of shapeless quasigroups.
- In 1999, Goldmann and Russell have shown that solving a system of equations in non-Abelian groups is an NP-complete problem.
- In 2001, Moore, Tesson and Therien have shown NP-completeness of the problem of solving a system of equations for even more general algebraic structures, i.e., monoids that are not a product of an Abelian group and a commutative idempotent monoid.

NTNU
Innovation and Creativity

Edon-*R* hash algorithm

- **Input:** (Q,*), N and M, where:
  - (Q,*) is a shapeless quasigroup of order 2^w, $w \geq 4$,
  - the number N is such that the length of the hash output is w x N bits and
  - M is the message to be hashed.
- **Output:** A hash of length w x N bits.

- **1. Pad** the message M, so the length of the padded message M' is multiple of N w-bit words i.e. $|M'|=k \times N$.
- **2. Initialize** $H_0=(0 \bmod 2^w, 1 \bmod 2^w, \ldots, 2N-1 \bmod 2^w)$.
- **3. Compute** the hash with the following iterative procedure:

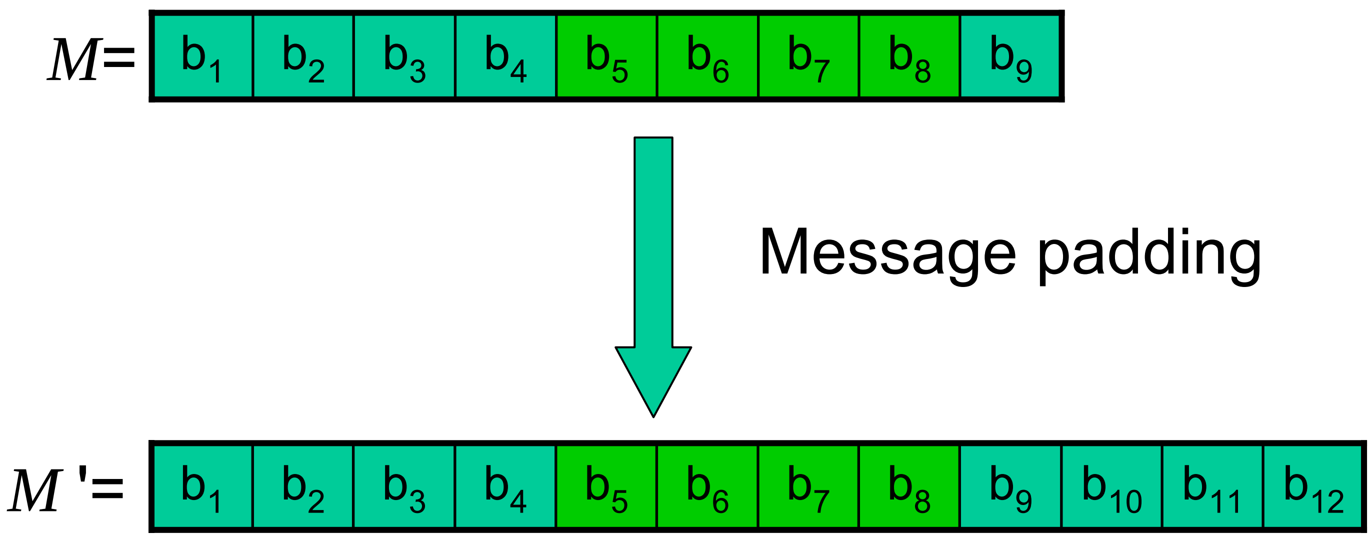
  For i=1 to k do

  $$H_i=R_1(H_{i-1} \,||\, M_i) \bmod 2^{2wN}$$
- **Output:**

  $$\text{Edon-}R(M)=H_k \bmod 2^{wN}$$

Edon-*R* hash algorithm (example)

N=4, Q={0,1,…,15}, (Q,*)



Edon-*R* hash algorithm (example)

$i=1$

R_1	0	1	2	3	4	5	6	7	b_1	b_2	b_3	b_4
b_4	.	.	.	.	.	.	.	.	.	.	.	.
b_3	.	.	.	.	.	.	.	.	.	.	.	.
b_2	.	.	.	.	.	.	.	.	.	.	.	.
b_1	.	.	.	.	.	.	.	.	.	.	.	.
7	.	.	.	.	.	.	.	.	.	.	.	.
6	.	.	.	.	.	.	.	.	.	.	.	.
5	.	.	.	.	.	.	.	.	.	.	.	.
4	.	.	.	.	.	.	.	.	.	.	.	.
3	.	.	.	.	.	.	.	.	.	.	.	.
2	.	.	.	.	.	.	.	.	.	.	.	.
1	.	.	.	.	.	.	.	.	.	.	.	.
0	h_1	h_2	h_3	h_4	h_5	h_6	h_7	h_8	h_9	h_{10}	h_{11}	h_{12}

NTNU Innovation and Creativity

Edon-*R* hash algorithm (example)

$i=2$

R_1	h_5	h_6	h_7	h_8	h_9	h_{10}	h_{11}	h_{12}	b_5	b_6	b_7	b_8
b_8	.	.	.	.	.	.	.	.	.	.	.	.
b_7	.	.	.	.	.	.	.	.	.	.	.	.
b_6	.	.	.	.	.	.	.	.	.	.	.	.
b_5	.	.	.	.	.	.	.	.	.	.	.	.
h_{12}	.	.	.	.	.	.	.	.	.	.	.	.
h_{11}	.	.	.	.	.	.	.	.	.	.	.	.
h_{10}	.	.	.	.	.	.	.	.	.	.	.	.
h_9	.	.	.	.	.	.	.	.	.	.	.	.
h_8	.	.	.	.	.	.	.	.	.	.	.	.
h_7	.	.	.	.	.	.	.	.	.	.	.	.
h_6	.	.	.	.	.	.	.	.	.	.	.	.
h_5	h_1	h_2	h_3	h_4	h_5	h_6	h_7	h_8	h_9	h_{10}	h_{11}	h_{12}

Q2S

NTNU
Innovation and Creativity

Edon-*R* hash algorithm (example)

$i=3$

R_1	h_5	h_6	h_7	h_8	h_9	h_{10}	h_{11}	h_{12}	b_9	b_{10}	b_{11}	b_{12}
b_{12}	.	.	.	.	.	.	.	.	.	.	.	.
b_{11}	.	.	.	.	.	.	.	.	.	.	.	.
b_{10}	.	.	.	.	.	.	.	.	.	.	.	.
b_9	.	.	.	.	.	.	.	.	.	.	.	.
h_{12}	.	.	.	.	.	.	.	.	.	.	.	.
h_{11}	.	.	.	.	.	.	.	.	.	.	.	.
h_{10}	.	.	.	.	.	.	.	.	.	.	.	.
h_9	.	.	.	.	.	.	.	.	.	.	.	.
h_8	.	.	.	.	.	.	.	.	.	.	.	.
h_7	.	.	.	.	.	.	.	.	.	.	.	.
h_6	.	.	.	.	.	.	.	.	.	.	.	.
h_5	h_1	h_2	h_3	h_4	h_5	h_6	h_7	h_8	h_9	h_{10}	h_{11}	h_{12}

NTNU Innovation and Creativity

Some properties of Edon-*R* family of hash functions

- Simple design.
- Edon-*R* is an infinite family of hash functions (there are about 2^{430} shapeless quasigroups of order 16, and MUCH MUCH MUCH more than 2^{192672} shapeless quasigroups of order 256).
- For every $N \geq 1$, the one-way compression function R_1 is a function from Q^{3N} to Q^{3N}, and from Theorem 1 and from the Hypothesis we can conjecture that finding preimages needs $\sim|Q|^N$ computational steps. Similarly Birthday attack is the best attack for finding collisions.
- One computation of the compression function needs $9N^2$ steps (quasigroup operations), but they can be easily parallelized and be executed in $6N$ steps.
- Internal parallelism of modern CPUs combined with the huge L1 cash that they have can be exploited for efficient implementations of the Edon-*R* family of hash functions. (Initial non-optimized C code achieves processing speeds 1 – 3 times faster then SHA-1 and SHA-2.)

Conclusions

- Edon-*R* is an infinite family of hash functions.
- It can be used for computing hashes of various lengths (80 bits, 128 bits, 160 bits, 192 bits, 1024 bits, 2048 bits, 20,000 bits, …).
- Its cryptographic strength relies on the one-way properties of the functions that are defined by transformations in shapeless quasigroups.
- Its internal memory is twice then the size of the hash output.
- It is conjectured that finding collisions or preimages for Edon-*R* functions is equivalent to solving a system of equations in shapeless quasigroups – and there is no mathematical knowledge or apparatus for efficiently solving such systems.
- It can be efficiently parallelized in hardware.
- It can be efficiently realized in software using the internal parallelism of modern CPUs and their L1 cash.

Q2S

NTNU
Innovation and Creativity

Thank you for your attention!