ERINDALE
family of hash functions

Nikolajs Volkovs

Dept. of Math. University of Toronto
Prata Technologies Inc.

Joint work with Kumar Murty

Comments from panel discussion at 2-d hash workshop

- **A.Shamir**: we need “something new” with a large internal state space
- **B.Praneel**: time to look at approaches different from Damgaard-Merkle
- **R.L.Rivest**: look at methods of working on the entire message rather than on a block-based procedure

ERINDALE
design feature

- It has a very large number of internal states (more than 2^50000)
- It is not based on Damgaard-Merkle structure
- It works with the entire message and it is very convenient for parallelization

ERINDALE
The idea of the construction

- It extracts features of different “sorts” from the message instead of “shaking” the bits
- The features are stored in special registers
- After finishing the process of extraction of the features from a message we start compressing the information that was collected in the registers
- The computation is a bit-stream procedure

ERINDALE
NIST's Randomness Tests

- For all the hash lengths specified by NIST (namely, 160, 224, 256, 384, 512 bits), the algorithm passed the randomness tests specified by NIST

ERINDALE
Performance results

- **Software implementation:**
- AMD Sempron 2GHz processor 3400+ using 1GB of RAM – close to SHA 384
- **Hardware implementation**
- On a Xilinx Virtex V FPGA at 299 MHz, we could reach **3.4 Gbps**
- SHA-512 at ~260 MHz runs at **~650 Mbps** (Xilinx Virtex XCV-1000-6)
- SHAvite-3 (Orr Dunkelman) on Virtex V **1.7 Mbps**

ERINDALE

- Family of functions with randomization and with a unique (size and value) padding for any message
- Has effective parameterization of:

  - security

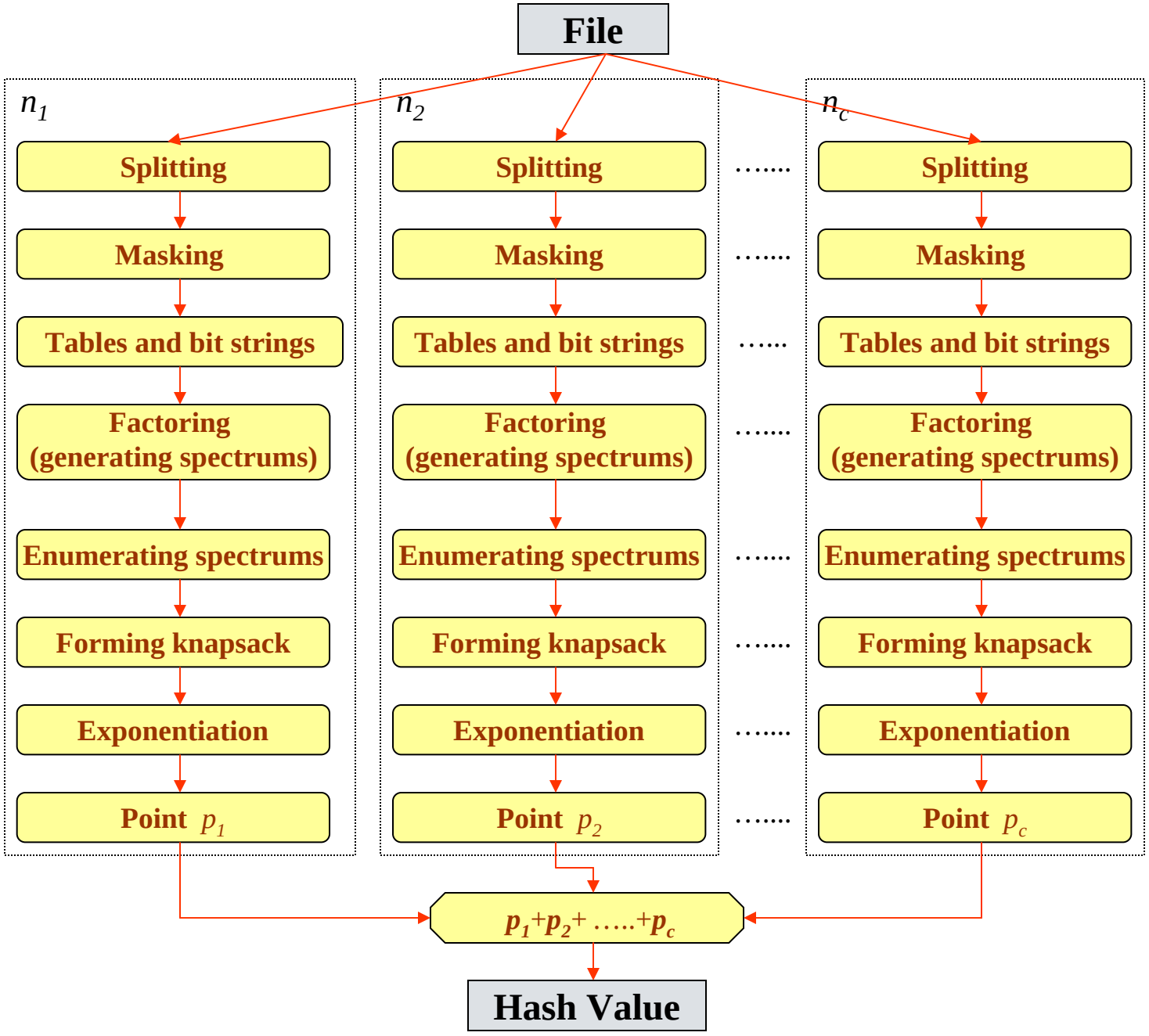
  - speed

  - “sorts” of features extracted from a message

  - size of a hash value

  - the inner structure

ERINDALE


File
n_1
n_2
n_c
Splitting
Masking
Tables and bit strings
Factoring (generating spectrums)
Enumerating spectrums
Forming knapsack
Exponentiation
Point p_1
Point p_2
Point p_c
.....
$p_1+p_2+ \ldots..+p_c$
Hash Value

Thank you.