School of Computing Science
Simon Fraser University, Canada

Analysis of Multimedia Authentication Schemes

Mohamed Hefeeda

(Joint work with KianooshMokhtarian)

12 May 2009

SFU

Motivations

- Increasing demand for multimedia services
- Content often transported over open and insecure networks (Internet)
- Many applications need to ensure authenticity of content
  - Videos for surveillance, documentary, political debates, etc
- Numerous authentication schemes exist
  - Merits and shortcomings against each other not clear
- No comprehensive analysis/comparison in literature

Our Work

- Define common performance metrics/scenarios
- Analytically analyze all schemes
- Conduct simulation and quantitative comparisons

- Recommendations for choosing appropriate scheme for a target environment
- Insights for further research

Outline

- **Performance Metrics**
- **(Brief) Overview of Authentication Schemes**
- **Analysis Results**
  - **Detailed derivations are given in the paper**
- **Conclusions and Recommendations**

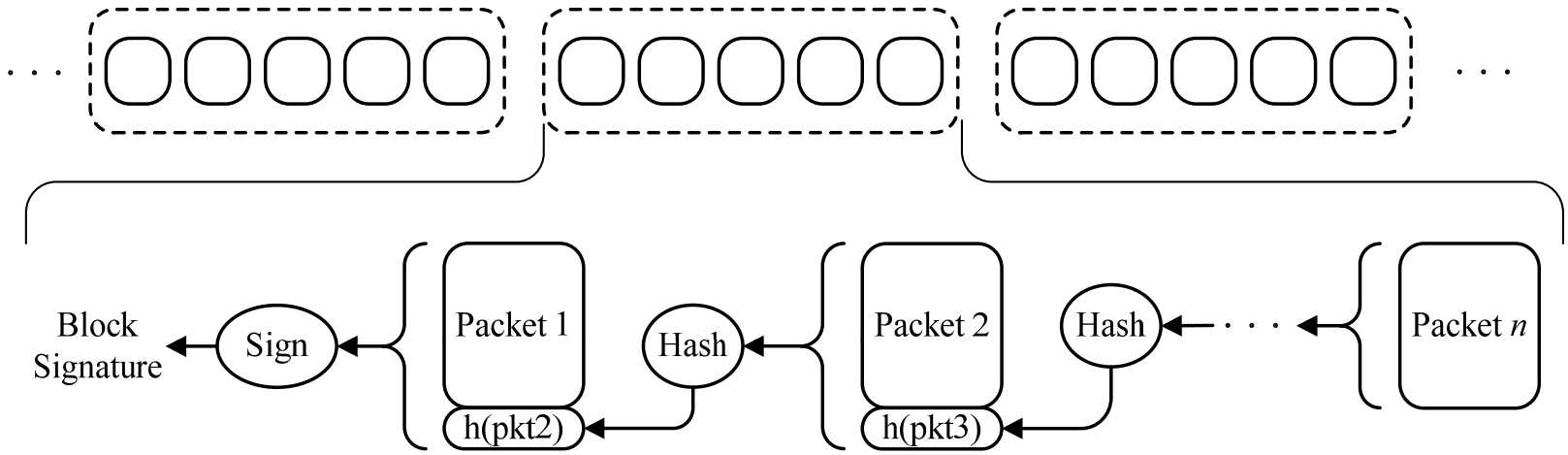
Performance Metrics

- **Computation cost**
  - Limited capacity receivers
- **Communication overhead**
  - Limited bandwidth
- **Tolerance against packet losses**
  - Bursty & random
- **Receiver buffer size required**
  - Memory constraints
- **Streaming delay**
  - Live streaming

Authentication Schemes for Videos

- **Present basic ideas of most important schemes**
- **Only representative sample**
  - **See our paper for details**

Hash Chaining [Gennaro 97]

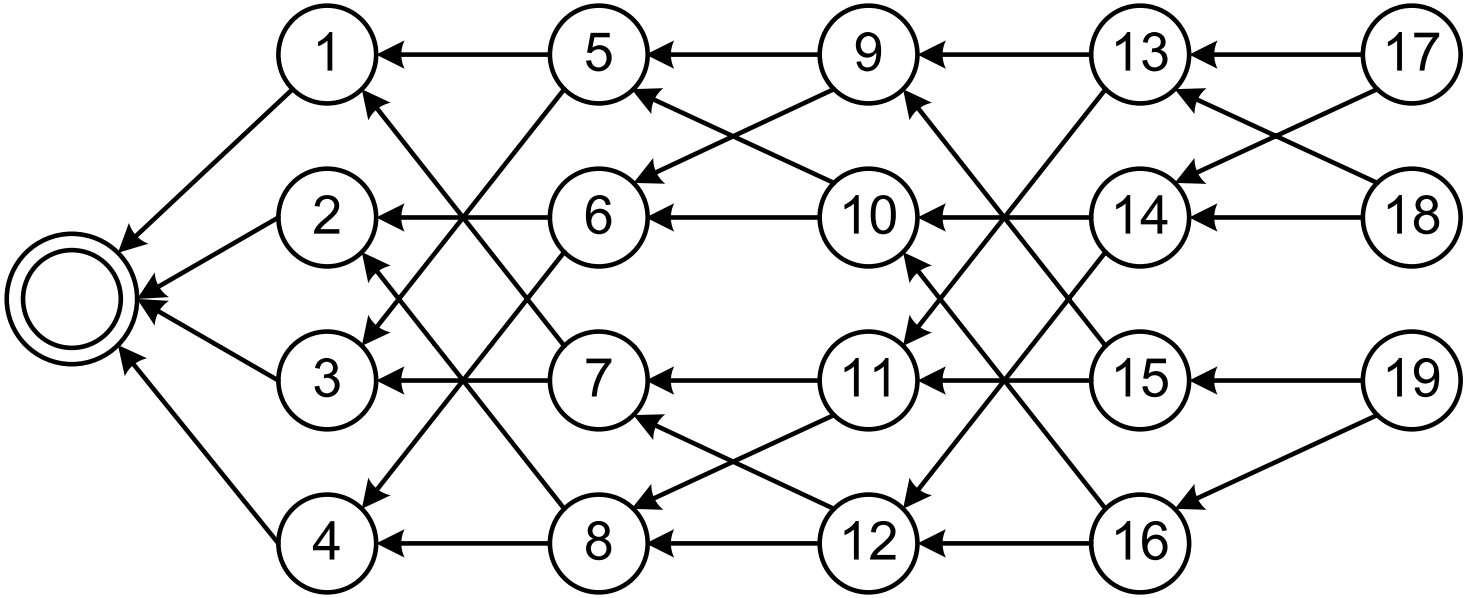


- **No receiver buffer required**
- **Delay: duration of a block (sender side) + zero (receiver side)**
- **No loss tolerance**

Hash Chaining: with Loss Tolerance

- Attach hash of packet to multiple other packets
- Choose other packets carefully
- Hashes of a block form Directed Acyclic Graph (DAG)
  - Packet is verifiable if it has path to signature packet

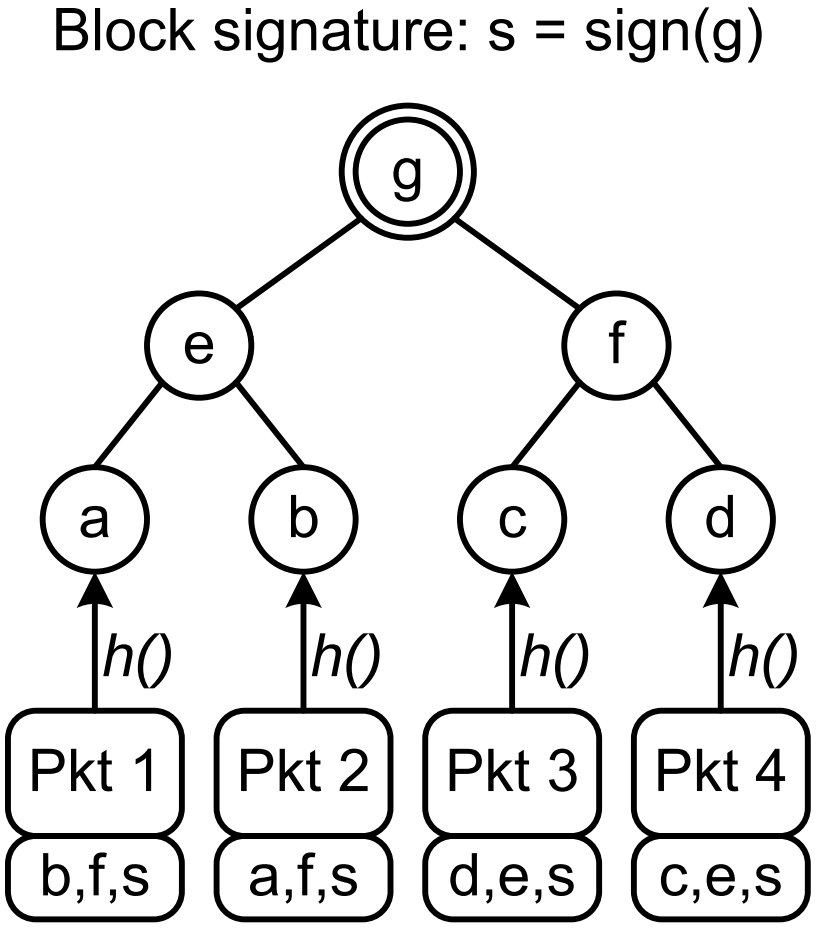
Butterfly Hash Chaining [Zhishou 07]



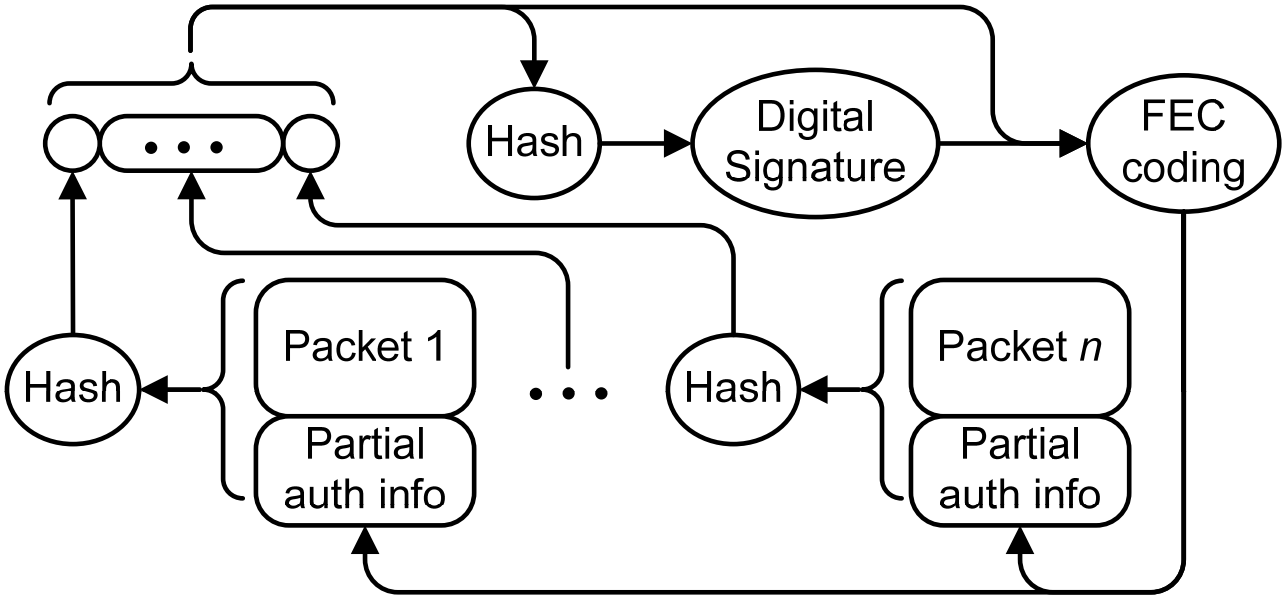
- **Improved loss tolerance**
- **Delay: duration of block (sender side) + zero (receiver side)**

Tree Chaining [Wong 99]

- **Based on Merkle hash tree**
- **Each packet carries all info needed for its verification**
  - **Complete loss tolerance**
- **No receiver buffer required**
- **Delay: duration of a block (sender side) + zero (receiver side)**



SAIDA: Signature Amortization using Information Dispersal Algorithm [Park 03]



- **Disperse auth info over *n* packets such that any *m* suffice to verify block**
  - ***m*: determines overhead—loss tolerance tradeoff**
- **Receiver buffer: *m*packets**
- **Delay: 2 times duration of block (sender + receiver side)**

SAIDA Improvements

- **eSAIDA (enhanced SAIDA) [Park 04]**
  - **To reducecomm overhead, one hash for each pair of packets**
    - **If packet is lost, its couple cannot be verified**
  - **A packet's hash is put in its couple with probability *s***
    - ***s* (input): determines overhead—loss tolerance tradeoff**
- **cSAIDA (communication overhead-reduced SAIDA) [Pannetrat 03]**
  - **A systematic FEC coding, keeping parity symbols only**
  - **An additional FEC coding**
- **Delay and receiver buffer of both: as in SAIDA**

TFDP: Tree-based Forward Digest Protocol [Habib 05]

- For streaming pre-encoded videos
- Similar to SAIDA, but block digests are not signed
  - Hash tree over block digests, root is signed
  - One signature for the whole stream
- Receiver buffer: nearly a complete block
- Delay: not relevant!

Sample of our Results

- **Analytic and numerical analysis**
- **Simulation analysis**

Computation Cost

Complete Table is given in the paper

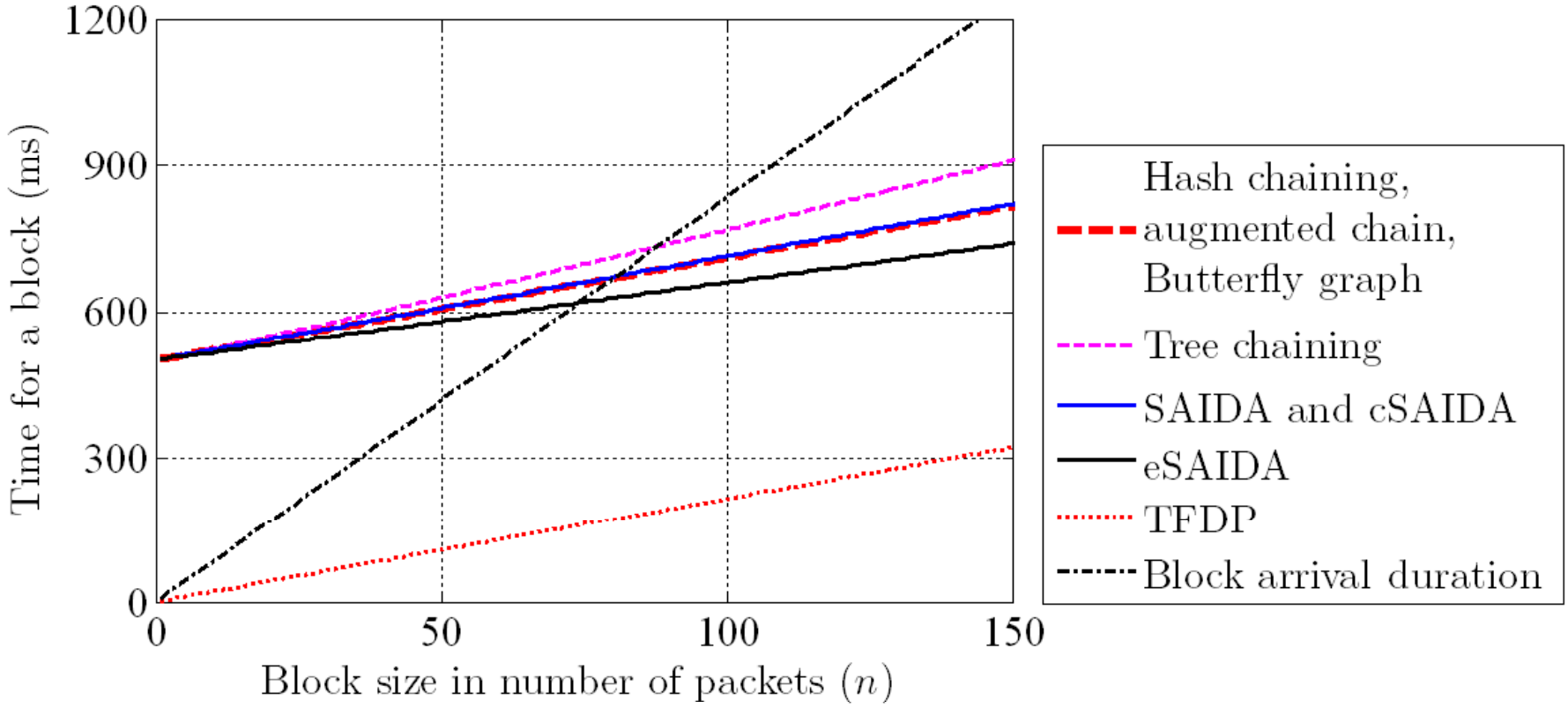
	num signature verif per block	num 512-bit hash operations per block
Hash Chaining	1	$n\lceil l/64 \rceil$
Tree Chaining	1	$n\lceil l/64 \rceil + (\lceil n \log n \rceil - n)\lceil s_{hash}/32 \rceil$
SAIDA	1	$n\lceil l/64 \rceil + \lceil s_{hash} n/64 \rceil$

n: block size

l: packet size

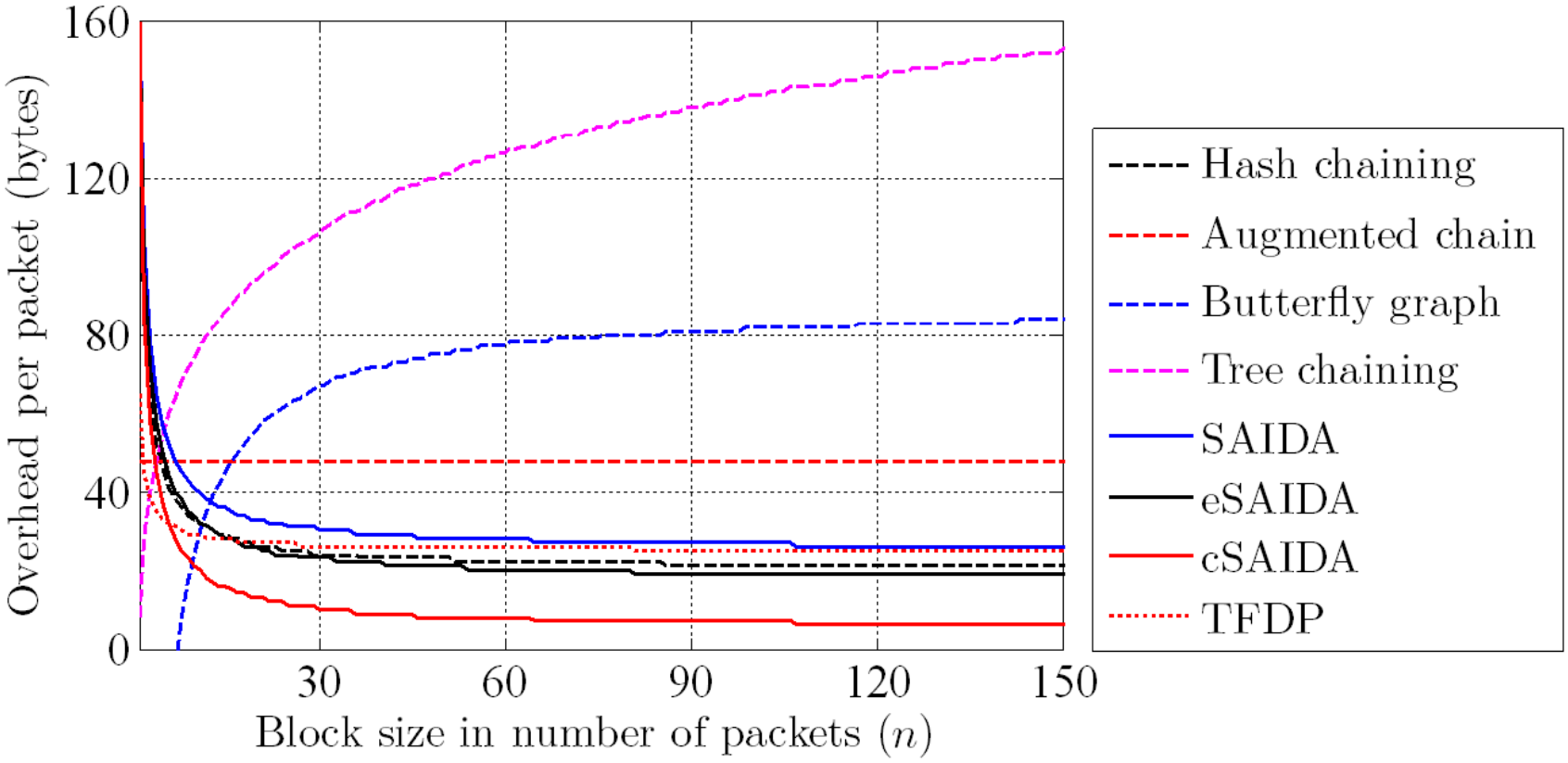
s_{hash}: hash size

Computation Cost



- **Time to verify block on limited-capability device➔ Lower bound on block size**

Communication Overhead



- **cSAIDA is the most efficient, Tree Chaining the least**

Delay and Buffer Requirements

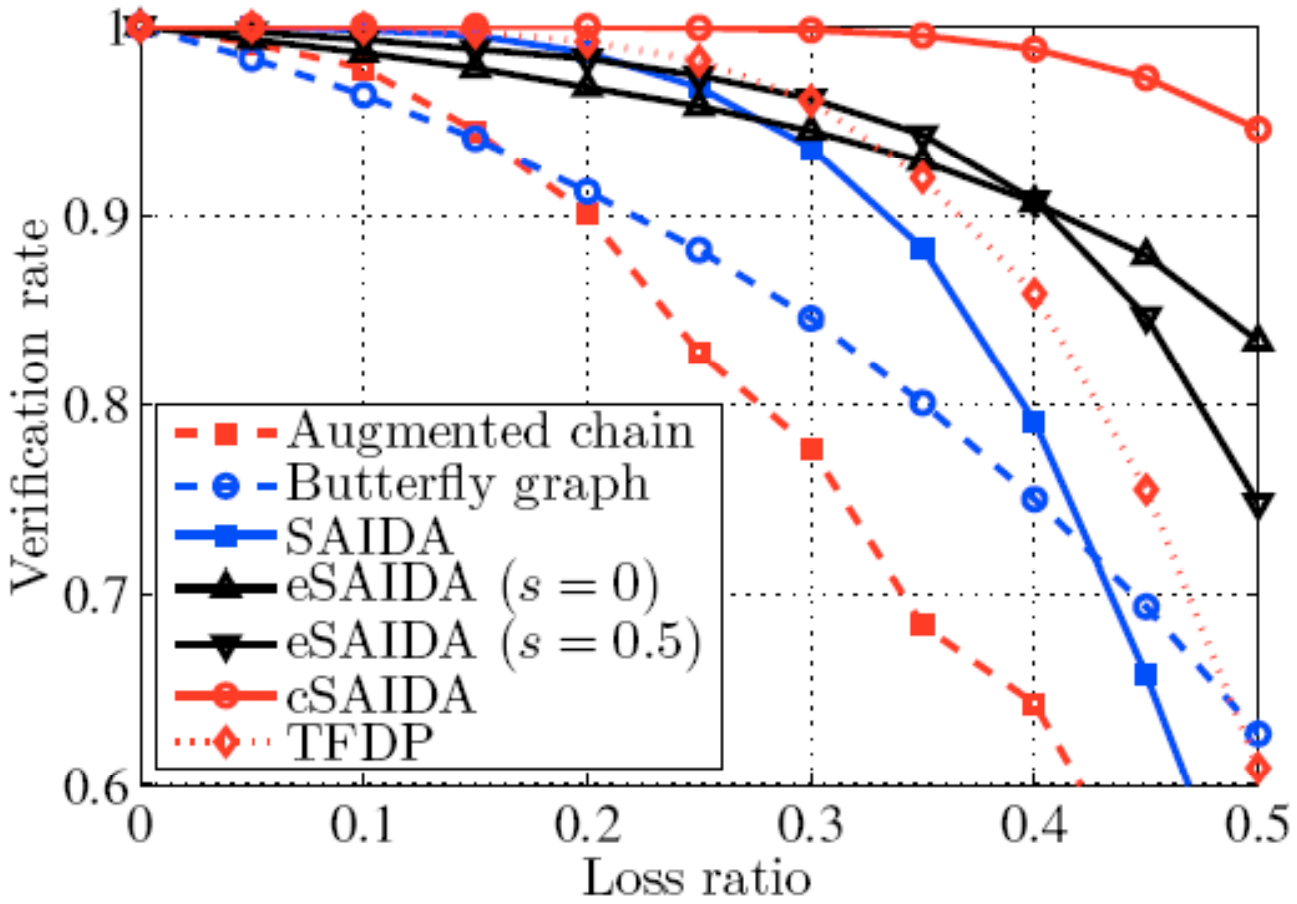
- **With a block size of 100 packets:**

	Buffer required (pkts)	Delay (seconds)
Hash Chaining	1	3-4
Augmented Chaining	n	3-4
Butterfly Chaining	n	3-4
Tree Chaining	1	3-4
SAIDA	n	6-7
eSAIDA	n	6-7
cSAIDA	n	6-7
TFDP	n	Not relevant

Simulation Results

- **Realistic parameter values from measurement studies**
- **Loss models**
  - **Bursty: Internet (router congestions)**
  - **Random: wireless networks and when interleaved packetization is used**
- **Best parameters are chosen for each scheme**

Simulation: Loss Resilience (Bursty Loss)



- **cSAIDA is the most efficient**

Conclusions

- **Conducted analytical and simulation comparisons among most authentication schemes for video streams**
- **Our Findings …**
- **Minimal computation cost**
  - **TFDP; for on-demand streaming only**
  - **Live streaming: all schemes almost the same**
- **Minimal communication overhead**
  - **cSAIDA**

Conclusions (cont'd)

- **Minimal delay**
  - **Hash/Augmented/Butterfly/Tree Chaining**
- **Maximal loss tolerance**
  - **Tree Chaining: high overhead, no buffering, low delay**
  - **cSAIDA: low overhead, but requires buffering one block, and incurs twice the delay of Tree Chaining**
- **Minimal buffering (memory) requirement**
  - **Hash Chaining; reliable data transfer only**
  - **Tree Chaining; fully loss tolerant**

Thank You!

Questions??

- **More info at:**

  http://nsl.cs.sfu.ca/