

Design and Implementation of a Tunable Encryption Service for Networked Applications

Stefan Lindskog and

Anna Brunstrom

Department of Computer
Science

Karlstad University, Sweden



Outline

- Background
- Objective
- Description of a tunable encryption (TE) service
- Evaluation and implementation
- Future work
- Conclusion

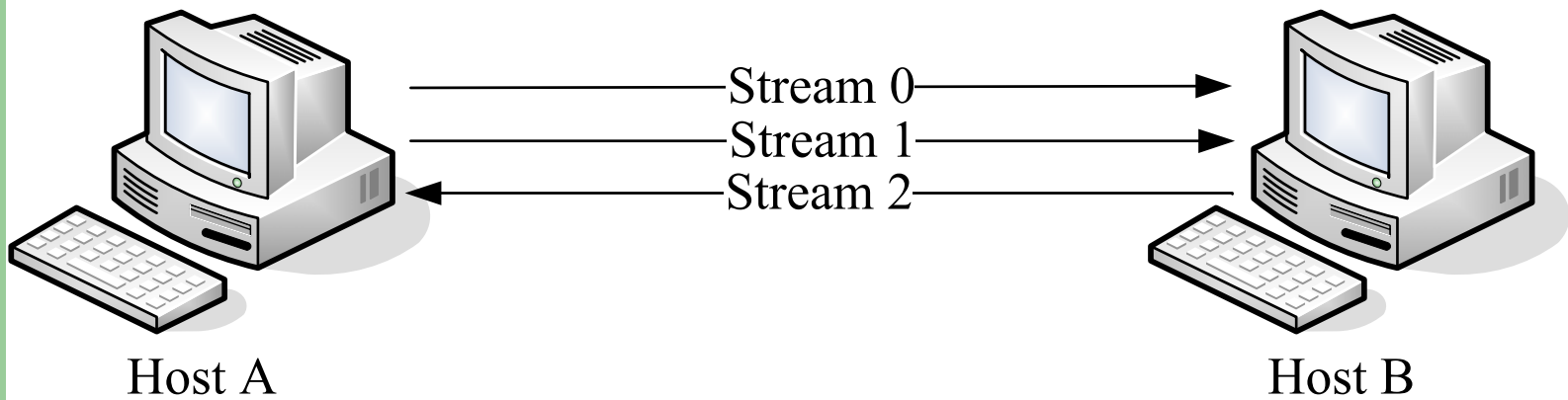
Background

- Network security is almost exclusively implemented through cryptography
- Cryptography gives adequate protection, but at high computational overhead (at the end points)
- Various lightweight security schemes have been proposed that reduce the computational overhead

Objective

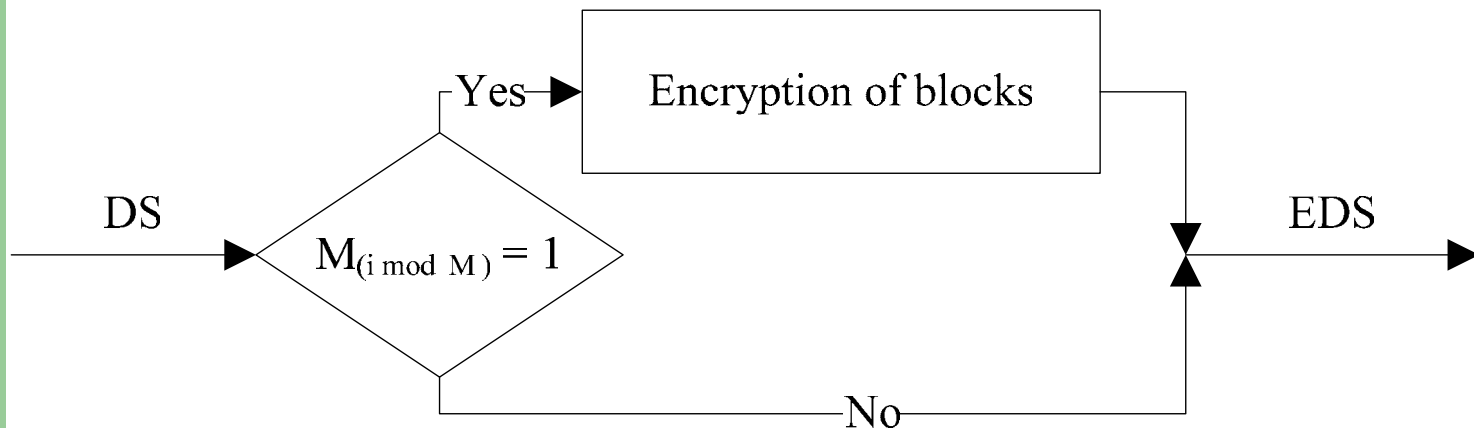
- We propose a TE service that aims to offer a tunable service with respect to QoS tradeoffs between computational cost and security
- The TE service is based on selective encryption and provides applications with a fine-grained control over which blocks to encrypt

Overview of the TE Service



Data Encryption

- The TE service provides selective encryption of blocks
- Data are encrypted using a symmetric block cipher
- Negotiation of encryption parameters is not part of the TE service



Specification of Encryption Level

- An encryption level (EL) is specified in an encryption control block (ECB)
- ECBs are transferred on stream 0
- An ECB consists of two fields:
 - a length indicator
 - a bit vector (encryption mask)
- Encryption masks are specified by applications

Analytical Evaluation

- An analytical evaluation of the computational gains has been conducted
- The estimated number of clock cycles (C) needed for encryption can be calculated using the formula:

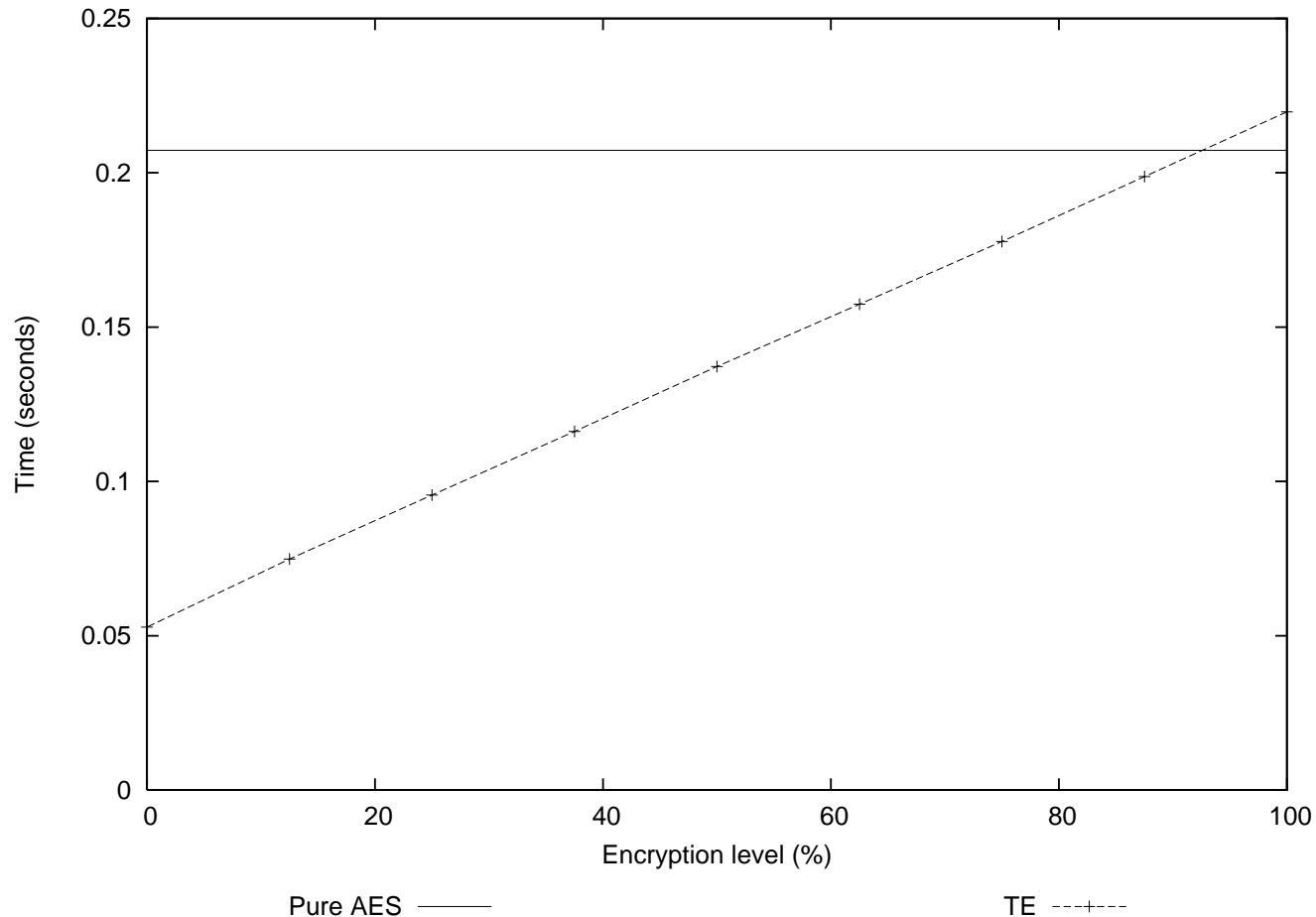
$$C = 17n + 440n_e + 32(n - n_e)$$

- Based on this formula, our TE service produces less overhead compared to encrypting everything (using ordinary encryption) when 96% or less of the content is encrypted

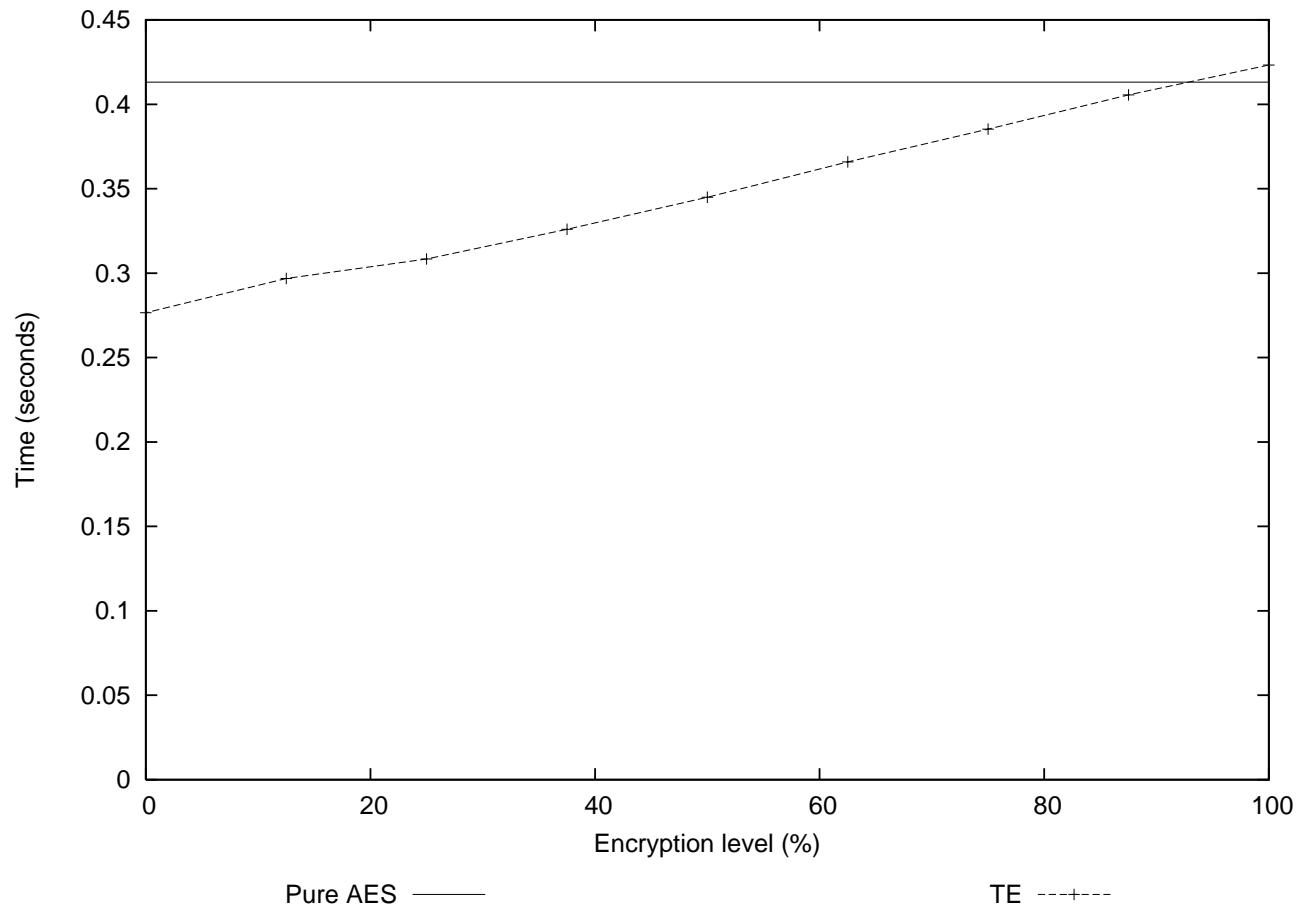
Implementation and Experimental Evaluation

- A prototype TE service has been implemented in the C/C++ programming language using SCTP [RFC 2960]
- AES was used as encryption algorithm
- Two PCs running FreeBSD were acting as sender and receiver
- Both PCs were configured with Gigabit Ethernet cards

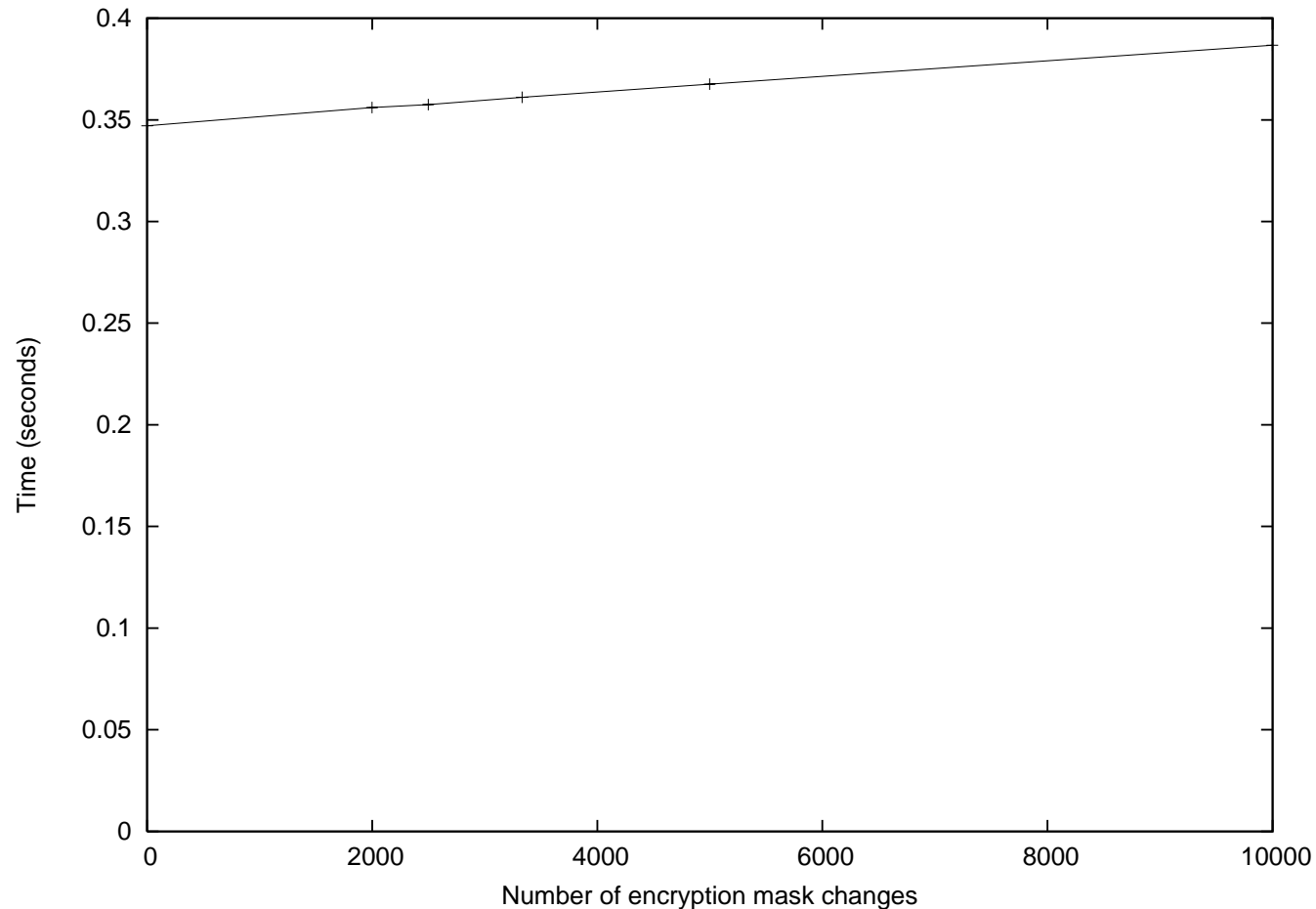
Measured Computational Gain (1)



Measured Computational Gain (2)



Impacts of Dynamic Changes



Future Work

- Investigation of how the TE service behaves at different ELs on an overloaded server
- Analysis of the achieved security at different ELs [QoP]
 - We are currently working on an idea based on guesswork
- Development of a TE middleware service

Concluding Remarks

- This paper presents a TE service that can be used by different applications and on different contents
- The proposed service is promising with respect to the potential reduction of computational overhead for encryption and can be used to provide QoS tradeoffs between performance and security



**“Good enough is good enough.
Perfect is too good.”**

[Bob Blakley]