Freie Universität Berlin

Performance and Security Tradeoff

Katinka Wolter

Bertinoro, June 26, 2010

Table of Contents

motivation

- what does the performance security tradeoff mean?
- we need to measure performance
- we need to measure security
- what are the costs of performance?
- what are the costs of security?
- can we trade one against the other?

performance

classical metrics

- throughput
- response time, completion time

evaluation tools

- CTMC
- queueing model
- GSPN, SRN, PEPA

measures

- accumulated reward
- expected reward
- moments of reward
- time to absorption

Performance versus Security

Quantification

- performance can be measured, quantified
- cost of performance can be quantified
- can we measure security?
- can we determine the cost of security?
- ultimately cost in terms of performance

Security Cost

It cost British Columbians almost $15 million a day to ensure a peaceful Olympics.

Members of the Vancouver 2010 Olympic Games Integrated Security Unit

Information Week

April 2007

- Forrester Research survey of 28 companies
- Security Breaches Cost $90 To $305 Per Lost Record
- 25% respondants do not know how to quantify loss

security cost

Google

Gmail now can be set to encrypt communications between a browser and Google's servers by default, an option that makes the e-mail service harder to snoop on but also potentially slower.

Google mail

Your computer has to do extra work to decrypt all that data, and encrypted data doesn't travel across the Internet as efficiently as unencrypted data, that's why we leave the choice up to you.

IBM slogans

IBM Security Solutions

Manage Risk. Reduce Costs. Enable Innovation.

IBM Virtualisation

Virtualisation Security Solutions from IBM Internet Security Systems^{TM}
Manage the risks of virtualisations and realise the cost savings.

IBM security

IBM cloud computing security

IBM offers end-to-end solutions that enable you to take a business-driven and holistic approach to securing your cloud computing environment. IBM's capabilities empower you to dynamically monitor and quantify security risks, enabling you to better:

- understand threats and vulnerabilities in terms of business impact,
- respond to security events with security controls that optimize business results,
- prioritize and balance your security investments.

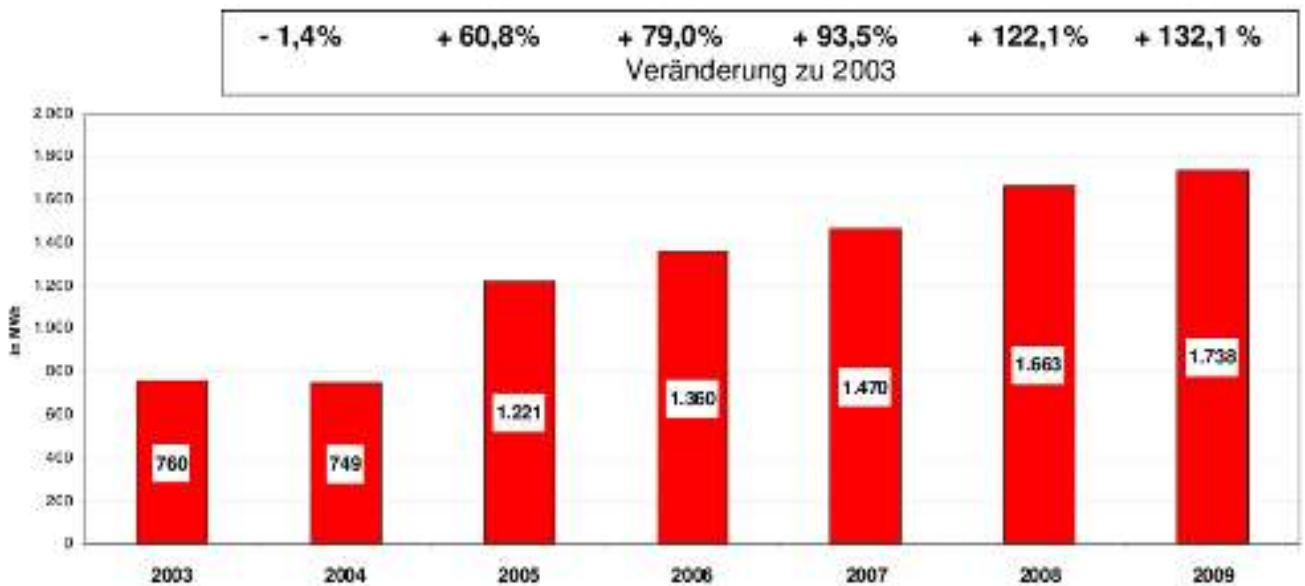
IBM Security Solutions for Data Centers

Your company can build a secure, dynamic information infrastructure that helps you accelerate innovation while reducing cost and complexity of security.

Freie Universität Berlin

IT costs

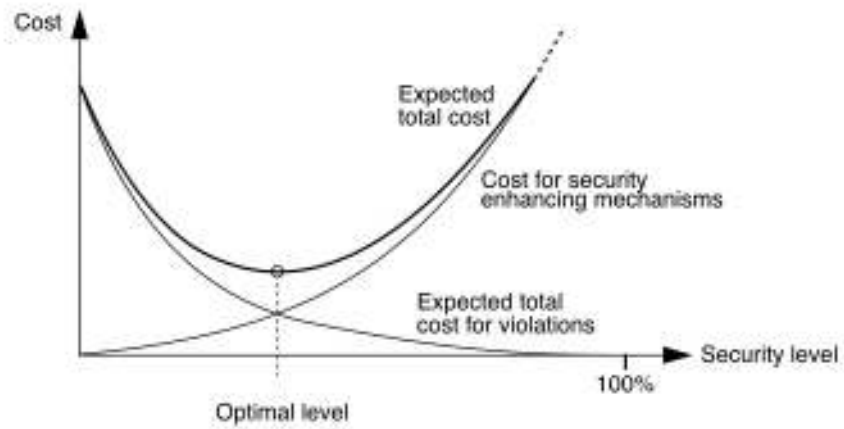
- total energy costs of FUB 10 M Euro
- electricity 50%
- power consumption of FUB's central IT services
- how much redundancy, security is necessary?



security concerns are not new

Problems

- cost of security incident unknown
- incidents may not be detected
- information security aims to get close to theoretical max. without knowing the cost.
- security risks may have very low probability. Don't invest close to potential damage to prevent, but detect.

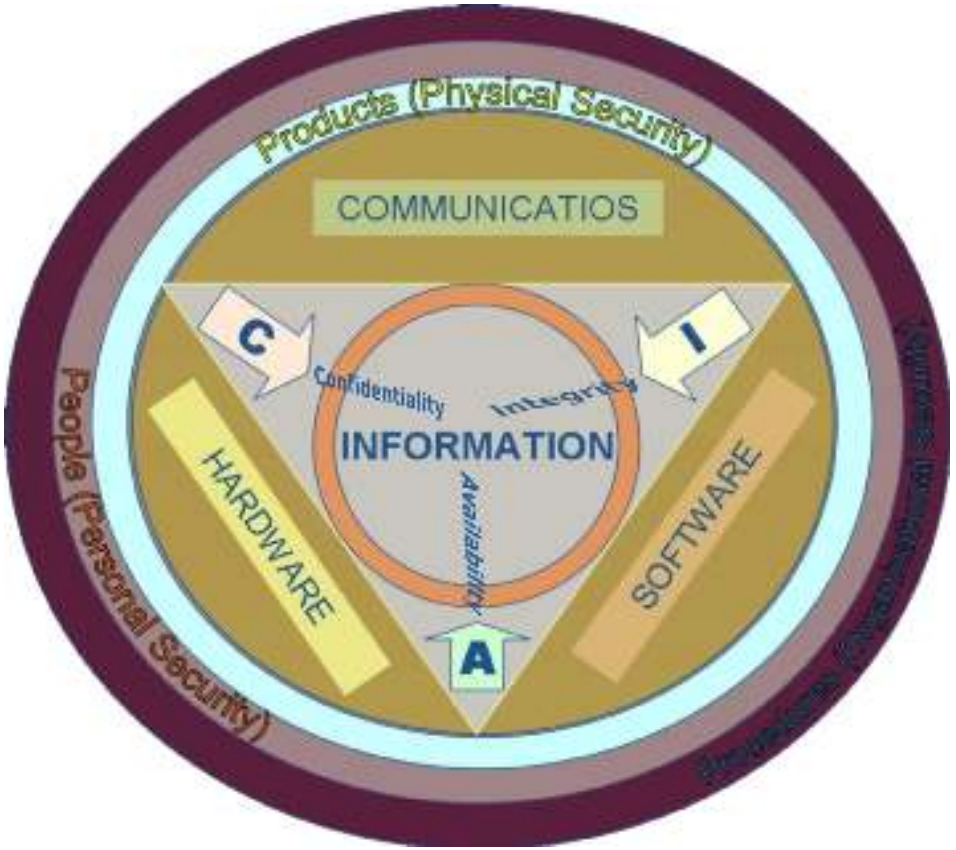


Source: A Structured Approach to Computer Security, T. Olovsson (1992)

Information Security

CIA Properties

- Confidentiality
  (information is not passed to unauthorised parties, defense)
- Integrity
  (information is not modified by unauthorised parties, banking)
- Availability
  (information is at disposition, telephone)
- (non-repudiation)
  sender and receiver are authentic



security versus dependability

analogies

- error, fault, failure in dependability
- vulnerability, security fault (Trojan hoarse), security failure
- failures can be modelled as random processes

differences

- accidental problems in dependability
- intentional problems in security
- attacker accumulates reward
- redundancy is helpful in dependability, detrimental for security

references

- Littlewood, Brocklehurst, Fenton, Mellor, Page, Wright (1993)
- Littlewood, Strigini (2004), Nicol, Sanders, Trivedi (2004)

weak hypothesis

survey of security quantification

- Verendel 2009: survey of 90 papers between 1981 and 2008.
- includes hardly model-based analysis
- it is unclear whether the methods applied are appropriate
- quantitative analysis needs large numbers of results
- solid, empirical data is necessary, hence

weak hypothesis

survey of security quantification

- Verendel 2009: survey of 90 papers between 1981 and 2008.
- includes hardly model-based analysis
- it is unclear whether the methods applied are appropriate
- quantitative analysis needs large numbers of results
- solid, empirical data is necessary, hence
- Quantified Security is a Weak Hypothesis

prevention

protect data and communication to avoid security breaches

diagnosis/detection

identify whether and when a security incident has happened

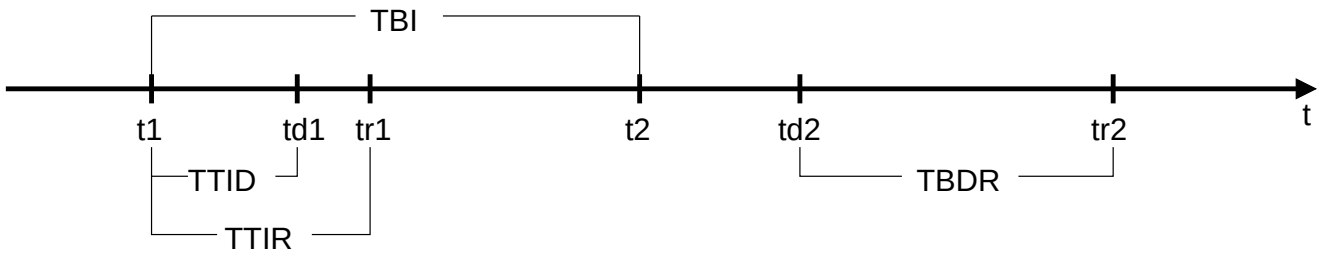
response

stop attack from causing further damage

recovery

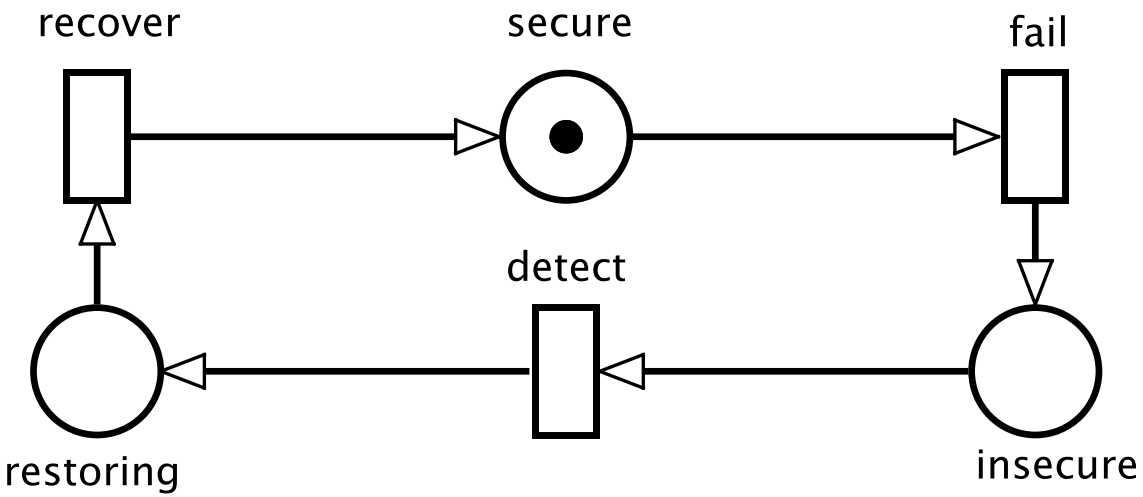
recover from security breach, rekey, use backup data

metrics for security in analogy with dependability metrics



- TBI: Time Between Incidents
- TTID: Time To Incident Discovery
- TTIR: Time To Incident Recovery
- TBDR: Time Between Detection and Recovery

simple Markovian security model



parameterise using

- inverse of MTBSI as rate of the *fail* transition
- inverse of MTTID as rate of the *detect* transition
- inverse of MTBDR as rate of the *recover* transition.

The states relate to prevention, diagnosis, recovery.
Open question: how do we know the rates?

performance cost of encryption

experiments

- experimental study, no model
- investigation of different algorithms for symmetric and asymmetric encryption
- investigation of different implementations
- encryption of 1,137 byte plaintext file
- keylength: DES 56bit, DESede (Triple DES) 112, Skipjack 80, 128 all others
- results for symmetric and asymmetric algorithms include key generation, algorithm initialization and message encryption times

C. Lamprecht, A. van Moorsel, P. Tomlinson, and N. Thomas. Investigating the efficiency of cryptographic algorithms in online transactions. *International Journal of Simulation: Systems, Science & Technology*, 7(2):63–75, 2006.

performance of Sun JCE implementation

- encryption times range between 85ms and 180ms
- triple DES (DESede) hardly slower than DES

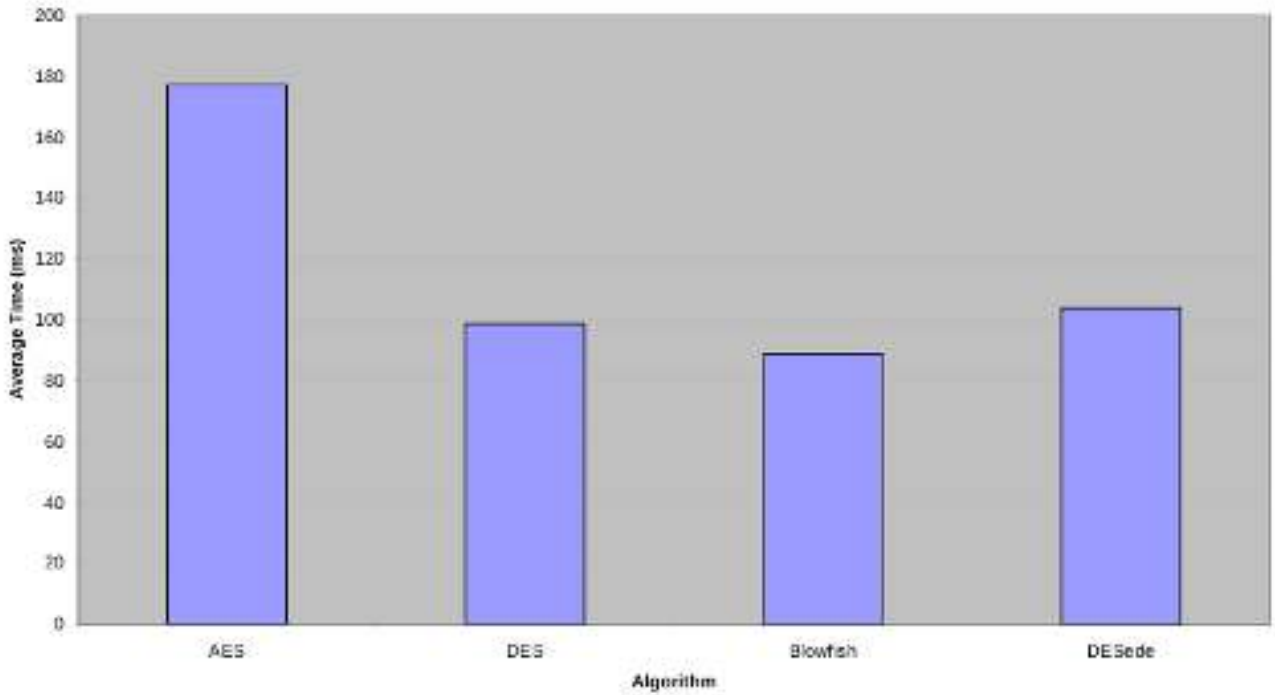


Figure 1: Average time to encrypt a 1137B file using JCE distributions

performance of Java Cryptix implementation

- encryption times range between 15ms and 50ms
- AES = Rijndael hardly slower than DES
- triple DES (DESede) slightly slower than DES

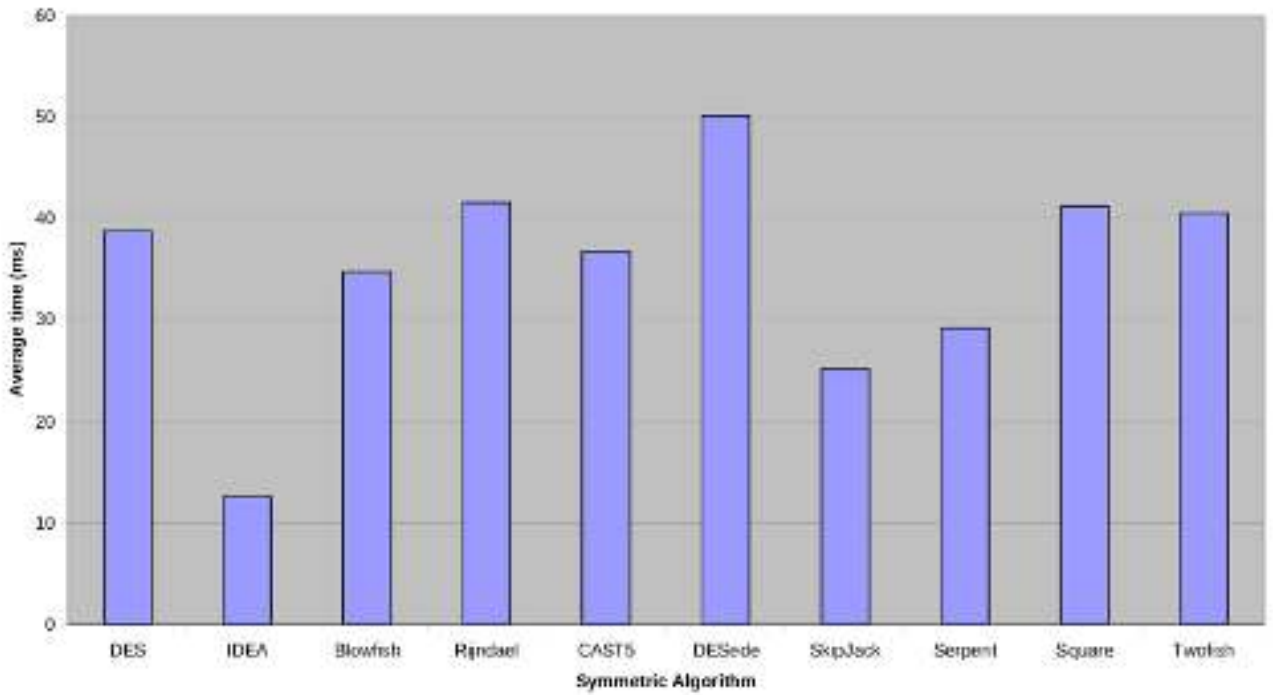


Figure 2: Average time to encrypt a 1137B file using Cryptix distributions

conclusions for symmetric encryption

performance versus security

- IDEA and Cryptix implementation seem to be best
- security measured in key length $\Rightarrow$ DES and Skypjack less secure
- security and cost do not correlate
- implementation matters

public key cryptography

- encrypt with destinations public key
- receiver decrypts with private key
- avoids problem of secure key transmission
- security increases with key length
- current security standard RSA-1024
- measurement of key generation and encryption time

speed of public key encryption

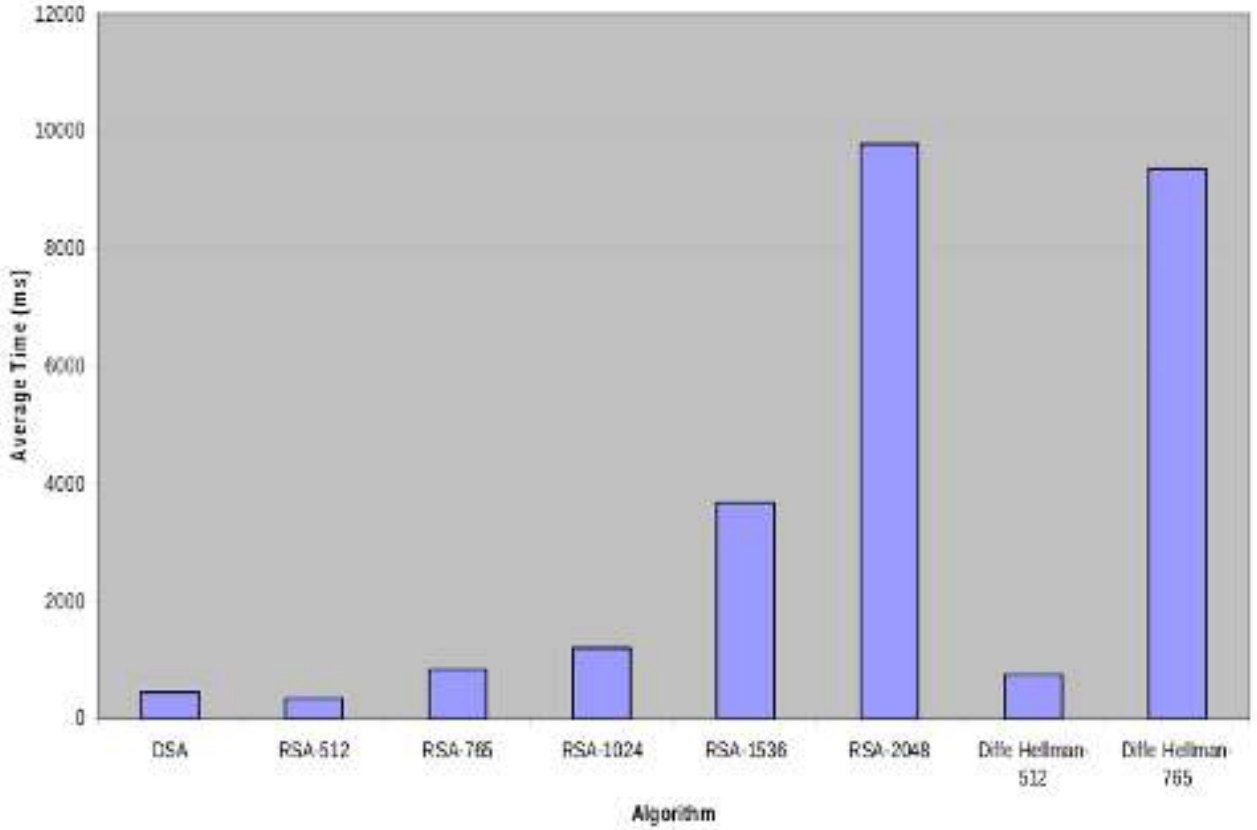


Figure 4: Average time for key generation and encryption using public key algorithms

- DSA only provides non-repudiation, no data confidentiality
- Diffie-Hellman 1024 is omitted for clarity

message digest

Cost of different algorithms to produce a message digest

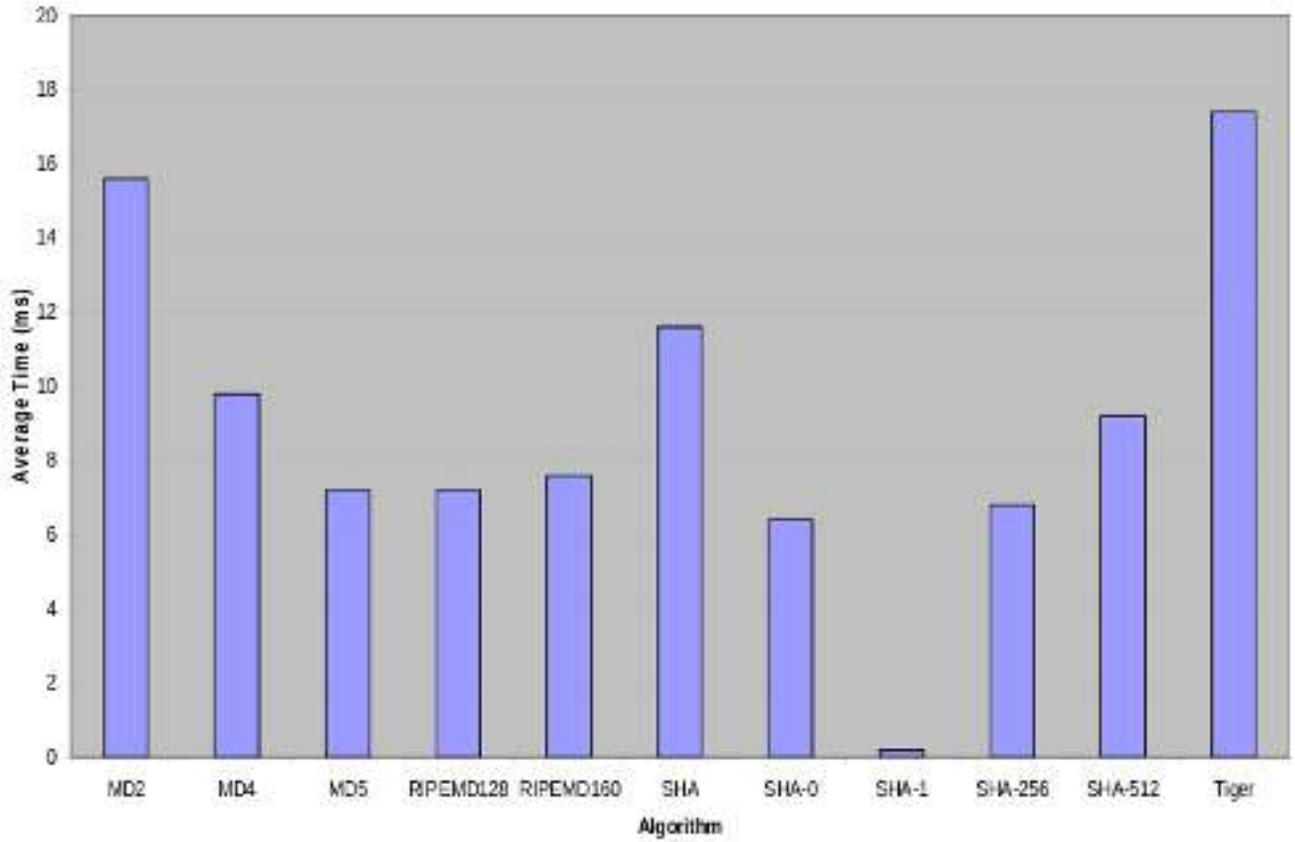


Figure 5: Average time to generate a message digest

summary encryption cost

symmetric encryption

IDEA is fastest

asymmetric encryption

best were:
RSA-1024 for public key encryption
SHA-256 for hashing (producing a digest)

performance security tradeoff

There is no indication that the recommendations provide a good tradeoff

performance of authentication algorithm

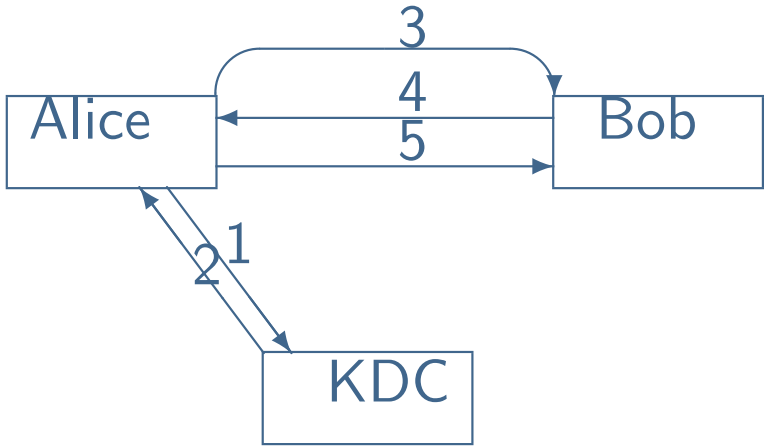
- key distribution for secure access to resources
- key distribution for secure communication
- stochastic process algebra model for the Needham-Schroeder protocol (Kerberos) from [Zhao&Thomas09]

questions

1. how many clients can a given KDC configuration support?
2. how much service capacity must we provide at a KDC to satisfy a given number of clients?
3. how long can a key be used before it is insecure?

Y. Zhao and N. Thomas, Efficient solutions of a PEPA model of a key distribution centre, Performance Evaluation, 67(2010), pp. 740–756

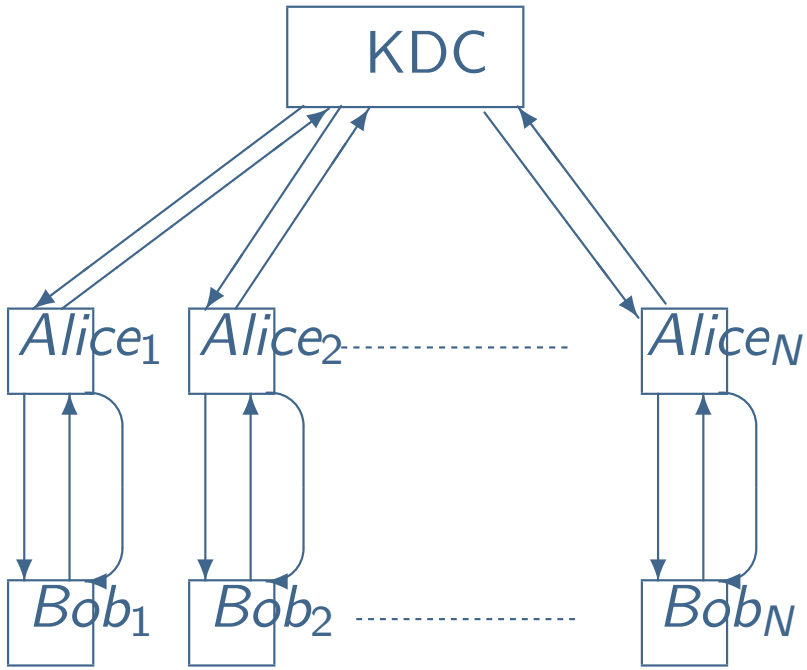
1. $Alice \longrightarrow KDC \ : \ A, B, N_1$
2. $KDC \longrightarrow Alice \ : \ \{K_S, A, B, N_1, \{K_S, ID_A\}_{K_B}\}_{K_A}$
3. $Alice \longrightarrow Bob \ : \ \{K_S, ID_A\}_{K_B}$
4. $Bob \longrightarrow Alice \ : \ \{N_2\}_{K_S}$
5. $Alice \longrightarrow Bob \ : \ \{f(N_2)\}_{K_S}$



- N_1 and N_2 are nonces (random items of data).
- ID_A is a unique identifier for Alice.
- $f(N)$ is a predefined function applied to the nonce N.

- Alice and KDC share a key K_A
- Bob and KDC share a key K_B

scalability



does it scale

modelling N pairs of Alice and Bob

PEPA model

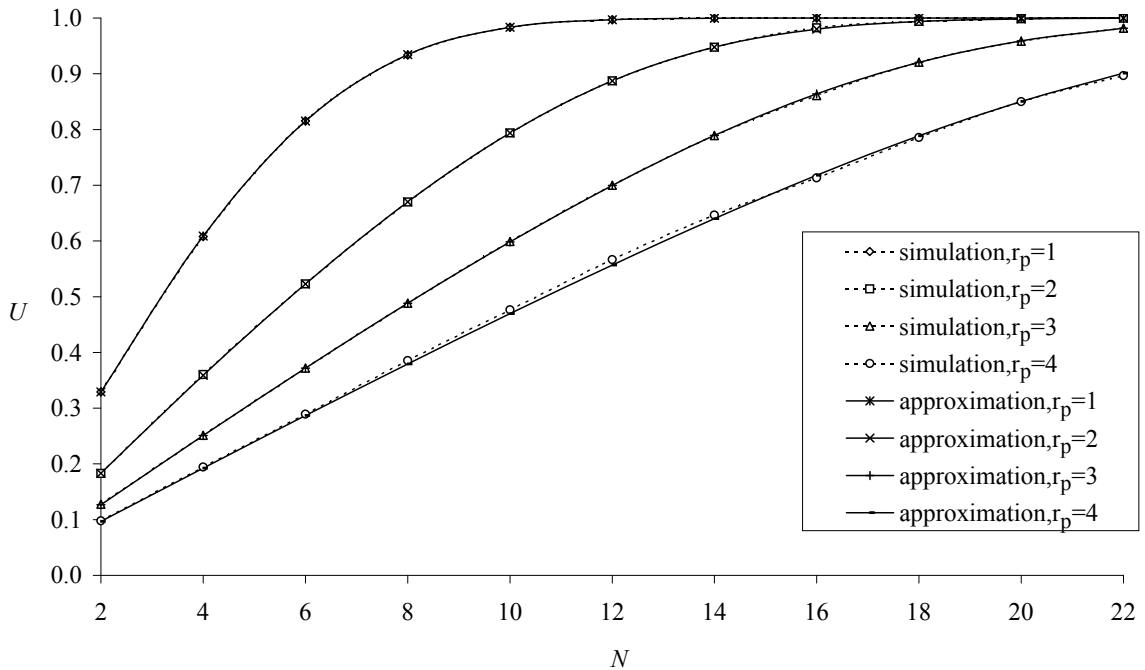
For $N = 1$

$$
\begin{aligned}
KDC &\stackrel{def}{=} (request, \top).(response, r_p).KDC \\
Alice &\stackrel{def}{=} (request, r_q).(response, \top).Alice' \\
Alice' &\stackrel{def}{=} (sendBob, r_B).(sendAlice, \top).(confirm, r_c).Alice'' \\
Alice'' &\stackrel{def}{=} (usekey, r_u).Alice \\
Bob &\stackrel{def}{=} (sendBob, \top).(sendAlice, r_A).(confirm, \top).Bob' \\
Bob' &\stackrel{def}{=} (usekey, \top).Bob \\
System &\stackrel{def}{=} KDC \underset{\mathcal{L}}{\bowtie} Alice \underset{\mathcal{K}}{\bowtie} Bob
\end{aligned}
$$

where, $\mathcal{L} = \{request, response\}$,
$\mathcal{K} = \{sendBob, sendAlice, confirm, usekey\}$.

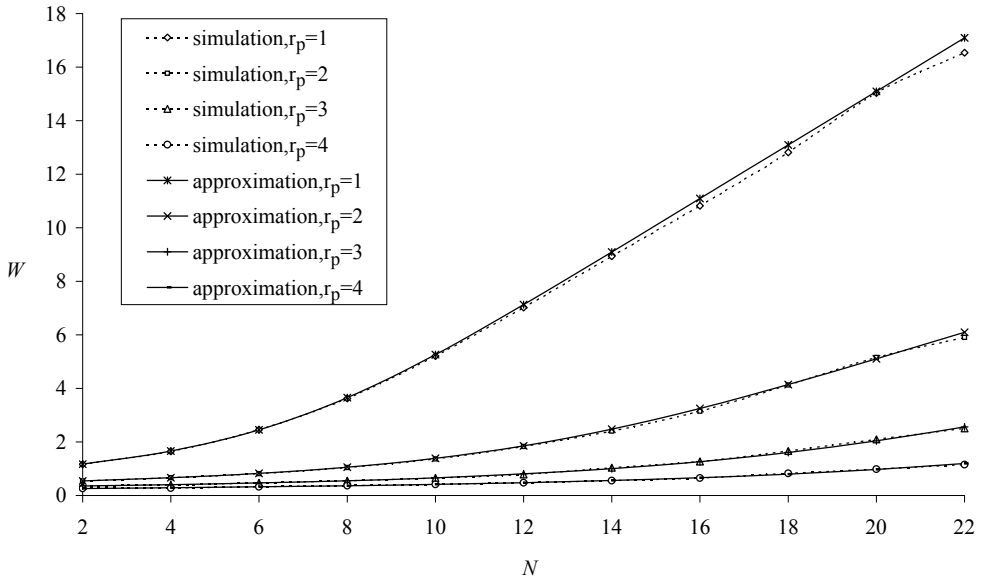
server utilisation of key distribution centre

a number of simplifications and approximations lead to results.



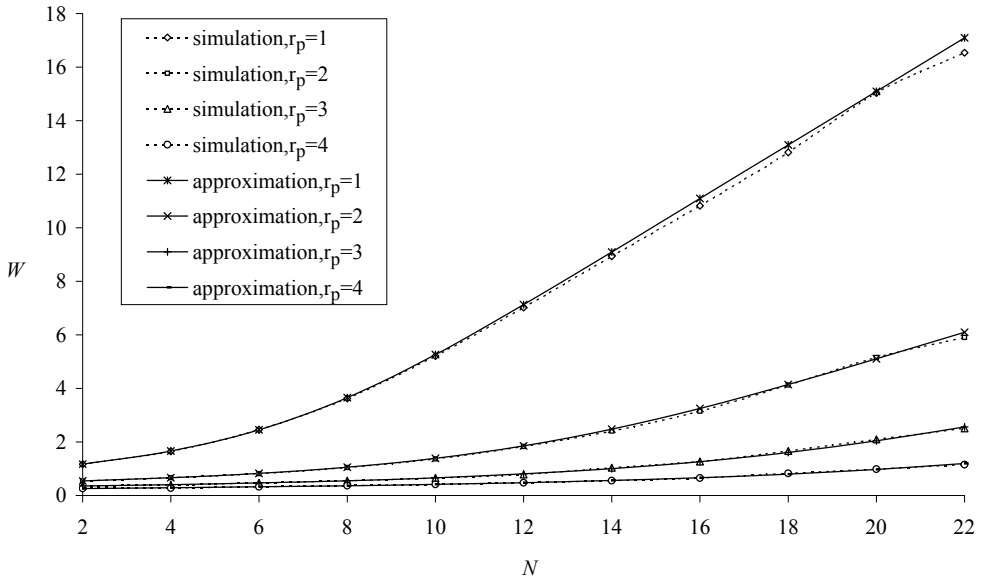
average utilisation versus the number of client pairs. $r_u = 1.1$, $r_A = r_B = r_c = r_q = 1$.

response time of key distribution centre



average response time versus the number of client pairs. $r_u = 1.1$, $r_A = r_B = r_c = r_q = 1$.

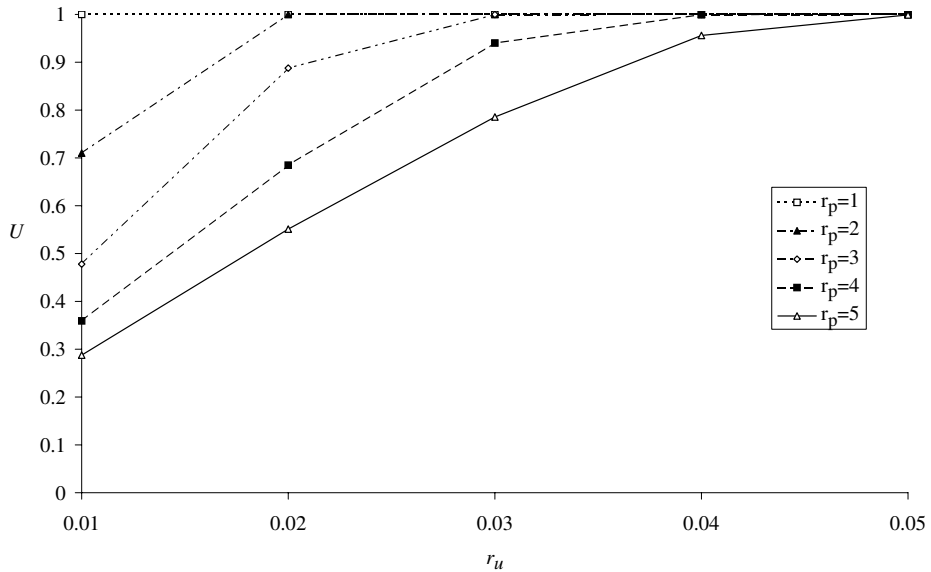
response time of key distribution centre



average response time versus the number of client pairs. $r_u = 1.1$, $r_A = r_B = r_c = r_q = 1$.

1. how many clients can a given KDC configuration support?
2. how much service capacity must we provide at a KDC to satisfy a given number of clients?

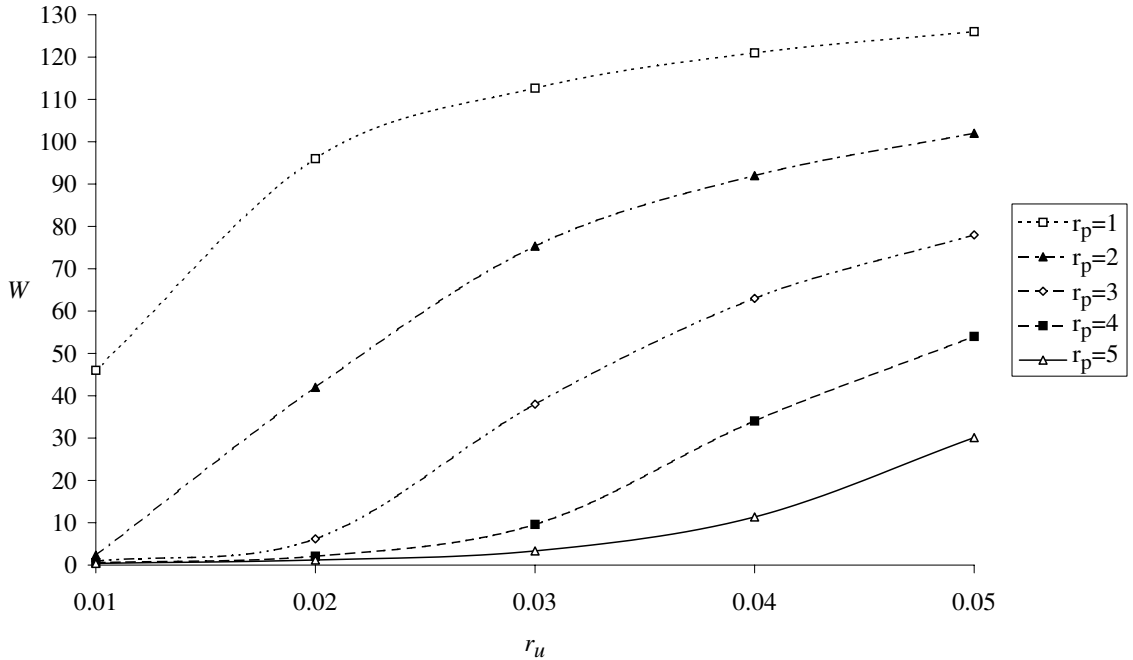
server utilisation of key distribution centre



average utilisation varied against the rate of session key use, r_u.

$r_q = r_A = r_B = r_c = 1$, $N = 150$.

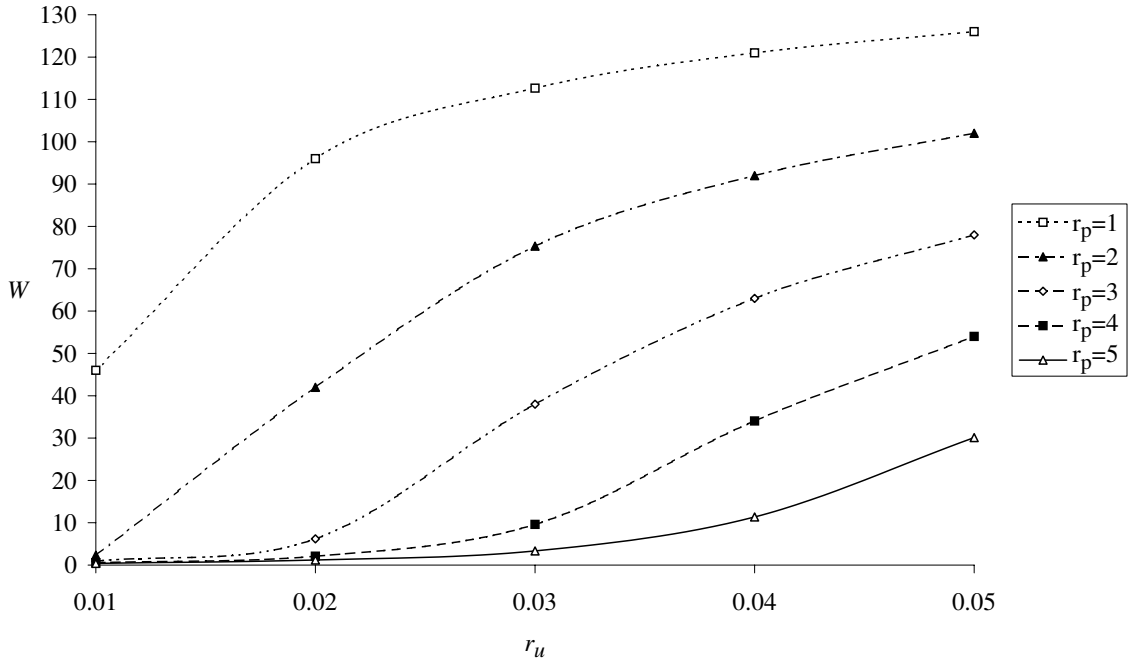
server response time of key distribution centre



average response time varied against the rate of session key use, r_u.

$r_q = r_A = r_B = r_c = 1$, $N = 150$.

server response time of key distribution centre



average response time varied against the rate of session key use, r_u.
$r_q = r_A = r_B = r_c = 1$, $N = 150$.

3. how long can a key be used before it is insecure?

Summary

- utilisation, response time of KDC increase with number of clients
- shorter use of session key increases security
- shorter use of session key increases utilisation and response time of KDC

but

- parameters do not translate to a system
- tradeoff between performance and security is not formulated

Freie Universität Berlin

Modelling Intrusion Tolerant Systems

security of intrusion tolerant system

- abstract model for system security
- purpose is to describe and quantify security
- compromise of confidentiality
- compromise of data integrity
- denial of service attacks
- description of security state
- stochastic process with levels of security

B. B. Madan, K. Goseva-Popstojanova, K. Vaidyanathan and K. S. Trivedi. A Method for Modeling and Quantifying the Security Attributes of Intrusion Tolerant Systems, Performance Evaluation (2004), 56, pp. 167–186.

states of the model

good state

preserved through

- authentication, access control, encryption
- firewalls, proxy servers
- strong configuration management, upgrades for known vulnerabilities

vulnerable state

reached through

- penetration
- exploration phases of an attack.

active attack state

- potential damage

more states

several degraded states

- masking through redundancy, backups (MC)
- restauration/reconfiguration possible (graceful degradation, GD) to handle DoS
- fail-secure to preserve confidentiality, integrity (FS)

several failed states

- intrusion detection fails (undetected compromised state, UC) (false negative)
- fail with alarm (F) (true positive)

counter measures

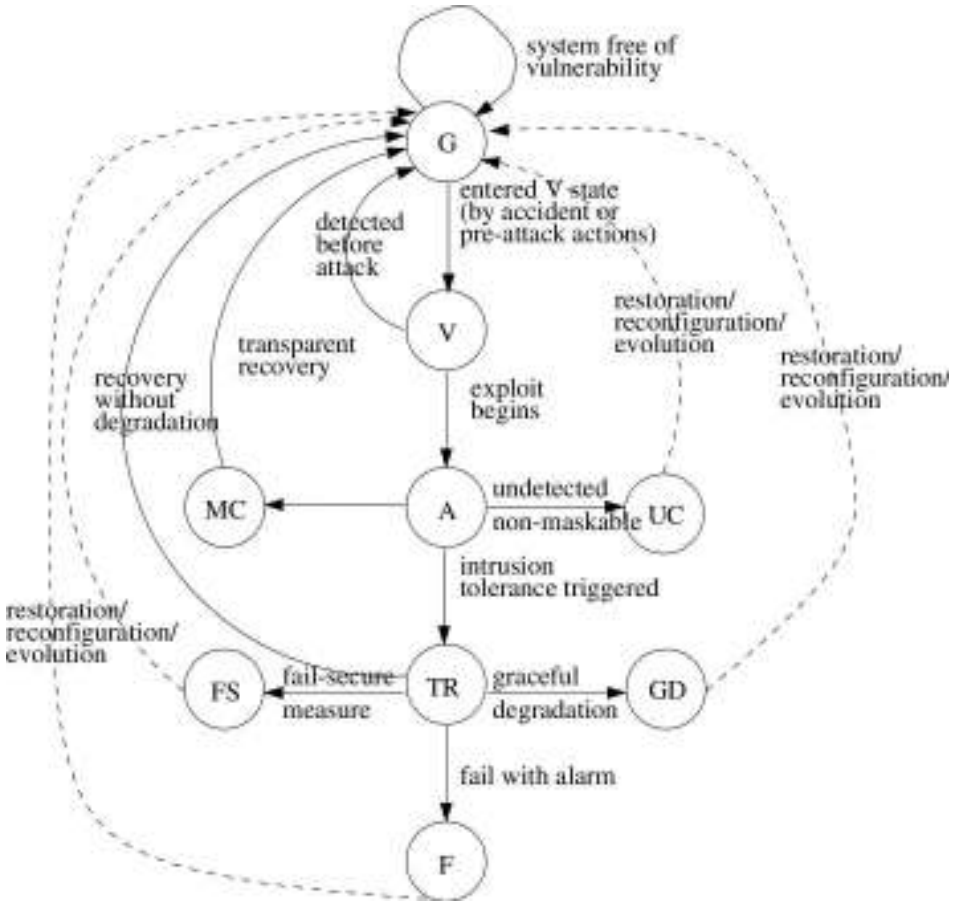
design and implementation of intrusion tolerant system

- error detection
- damage assessment
- error recovery, updates (redundancy)
- fault treatment

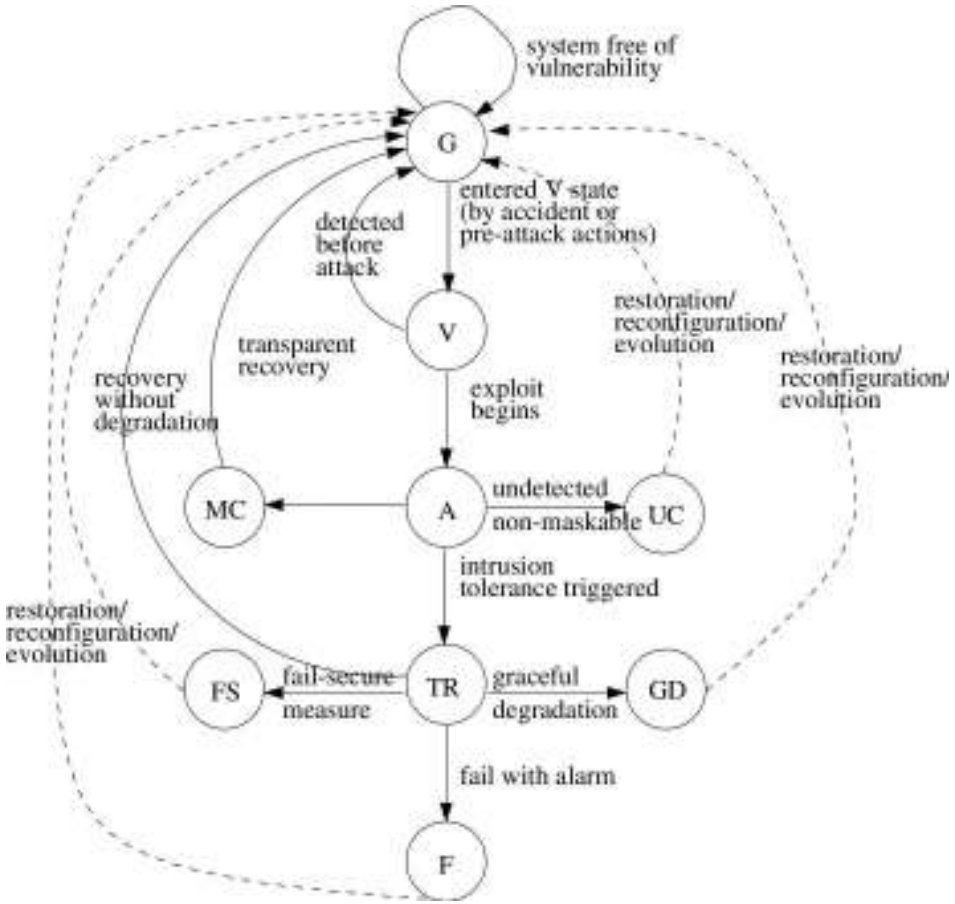
recovery states

- graceful degradation prevents denial-of-service attack
- stop system to protect confidentiality or data integrity

state-transition model



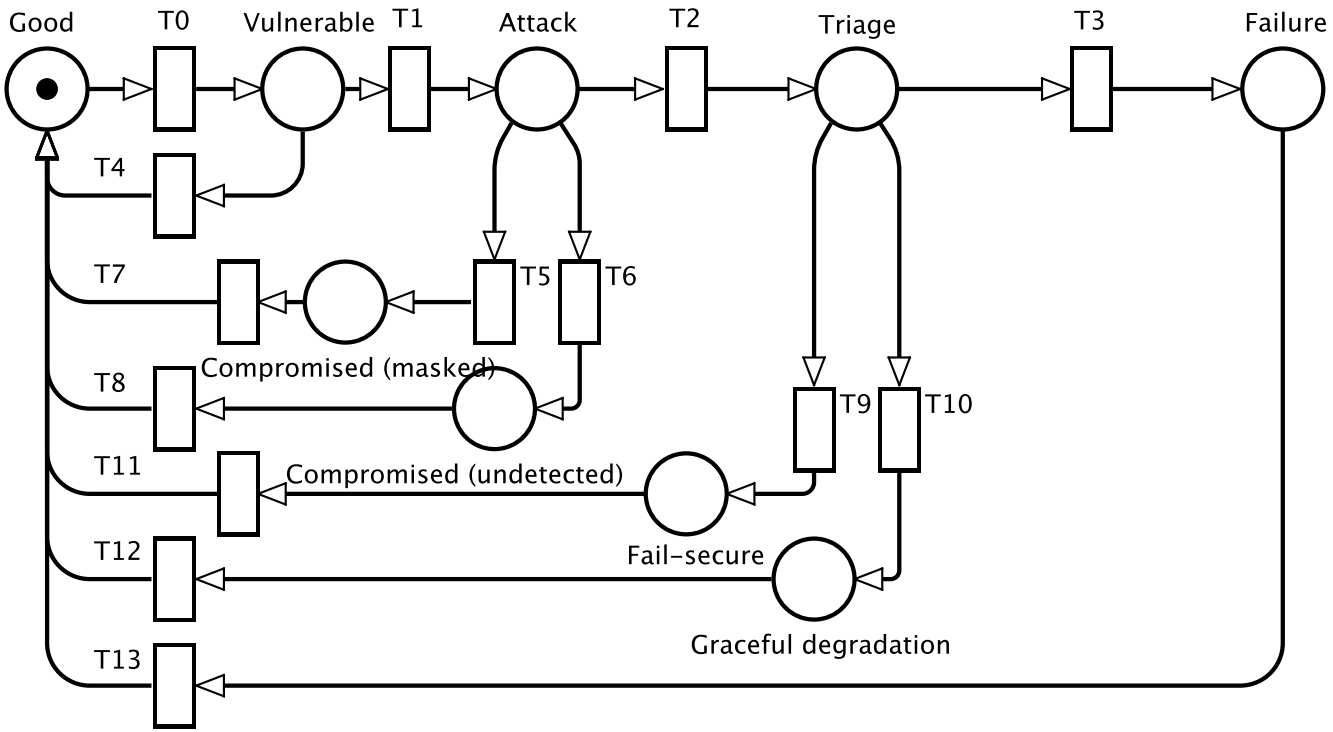
state-transition model



possible outcome of analysis

where should I invest, depending on attack model?

GSPN model



- unavailable in states FS, F, UC, $A = 1 - \pi_{FS} - \pi_F - \pi_{UC}$
- for DoS, $A_{DoS} = 1 - (\pi_F + \pi_{UC})$
- for MTTSF states UC, GD, FS, F are absorbing states, compute time to absorption in a DTMC.

measures

considered measures

- availability
- mean time to security failure (MTTSF)

parameters

- mean sojourn times
  $h_g = 1/2, h_V = 1/3, h_A = 1/4, h_{MC} = 1/4, h_{UC} = 1/2, h_{TR} = 1/6.$
- p_a probability of successful attack from vulnerable state

results: availability

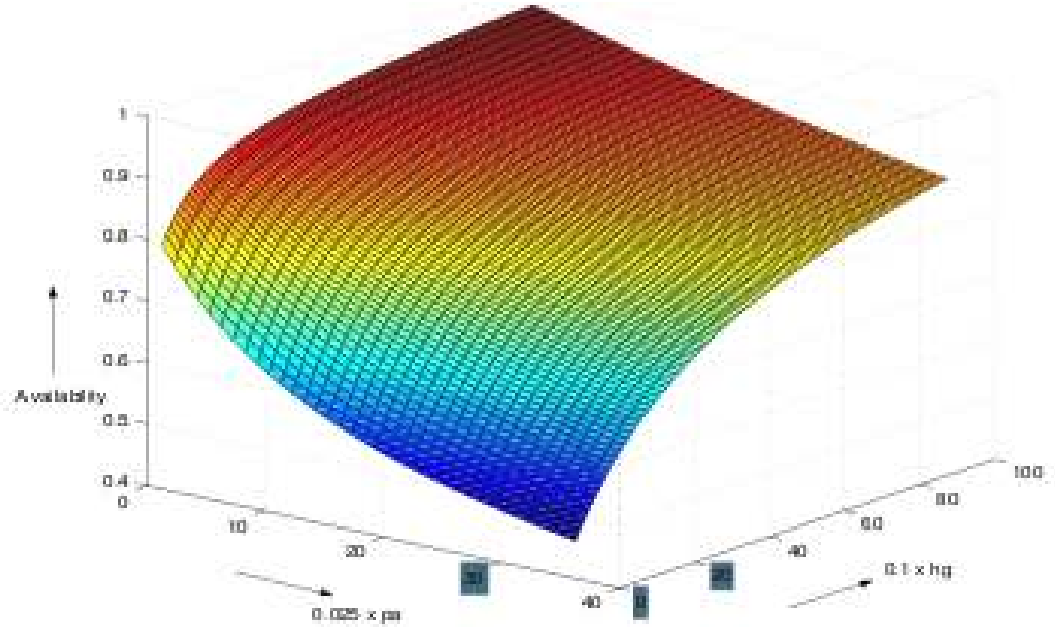


Figure 6. Availability as a function of p_a and h_G

insights

- higher probability of successful attack from vulnerable state p_a reduces availability
- longer mean time in the good state h_G increases availability

results: mean time to security failure

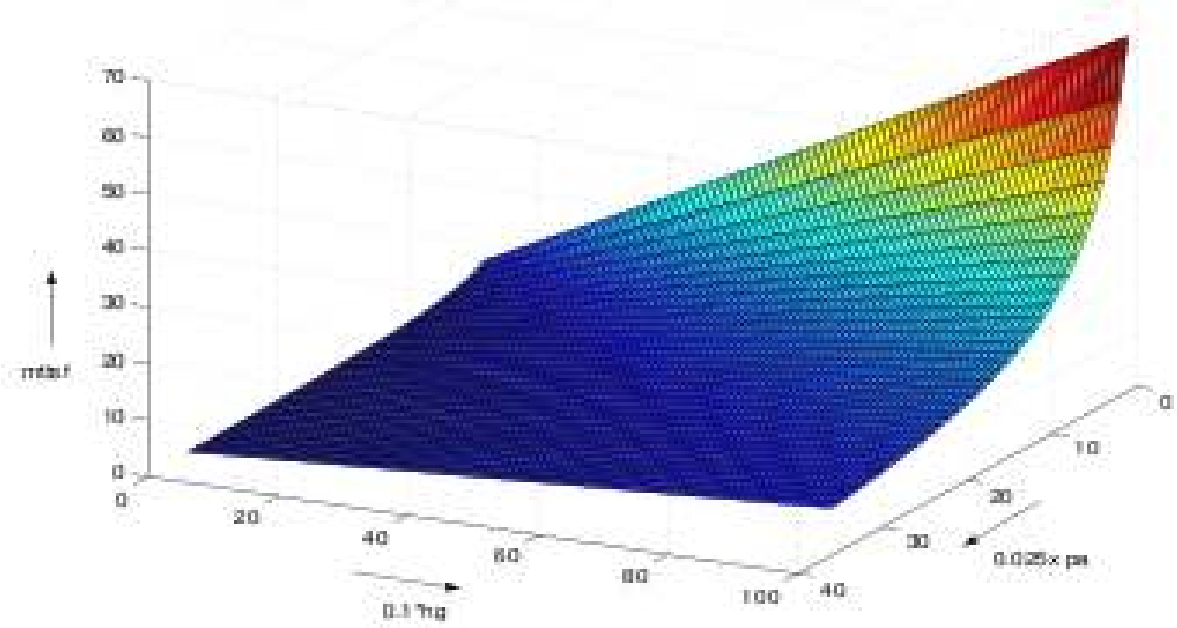


Figure 5. MTTSF as a function of p_a and h_G

insights

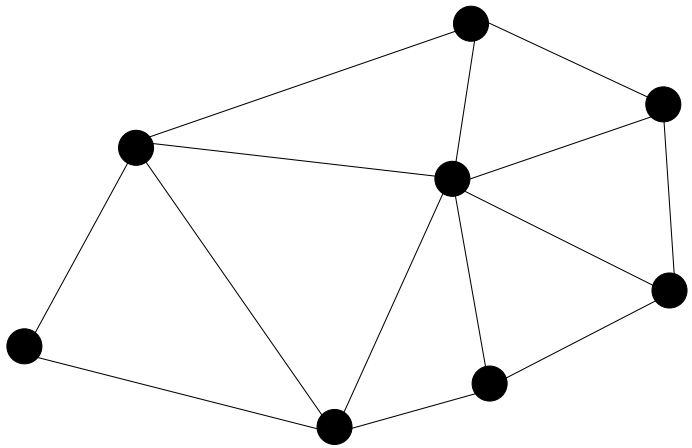
- MTTSF increases with longer mean time in the good state h_G
- MTTSF decreases with higher probability of successful attack from vulnerable state p_a.

modelling an intrusion tolerant system

- flexibel model that can represent different types of attacks
- quantification of security (considering DoS, confidentiality, integrity attacks)
- inspired by performability analysis
- doubtful parameter choices (planned improvements using SITAR)
- no notion of performance (planned improvements)
- no security cost
- no tradeoff

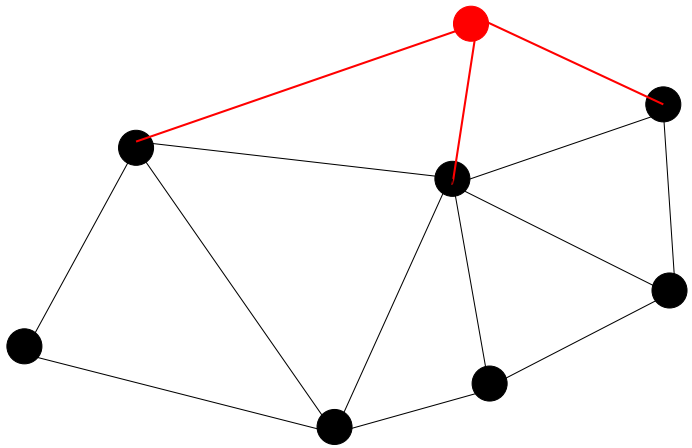
security of MANETs

- group communication in mobile ad hoc network using group key
- intrusion detection system (IDS) checks for compromised nodes



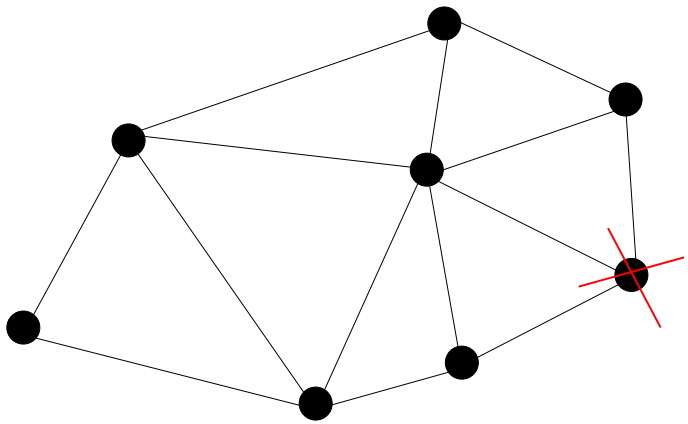
security of MANETs

- group communication in mobile ad hoc network using group key
- intrusion detection system (IDS) checks for compromised nodes
- IDS may not detect (false negative)



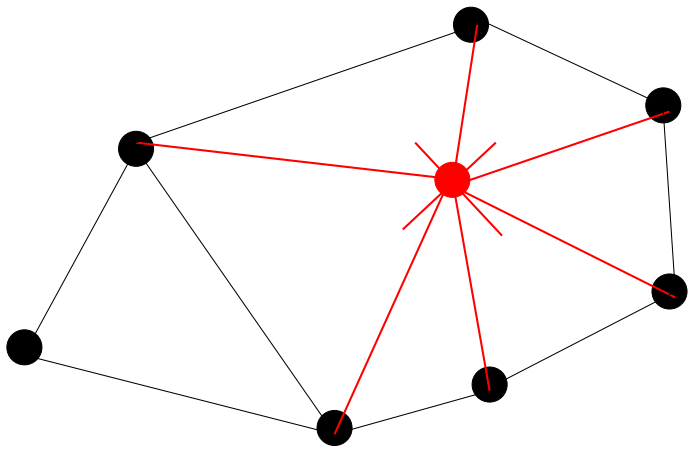
security of MANETs

- group communication in mobile ad hoc network using group key
- intrusion detection system (IDS) checks for compromised nodes
- IDS may not detect (false negative)
- IDS may erroneously detect (false positive)



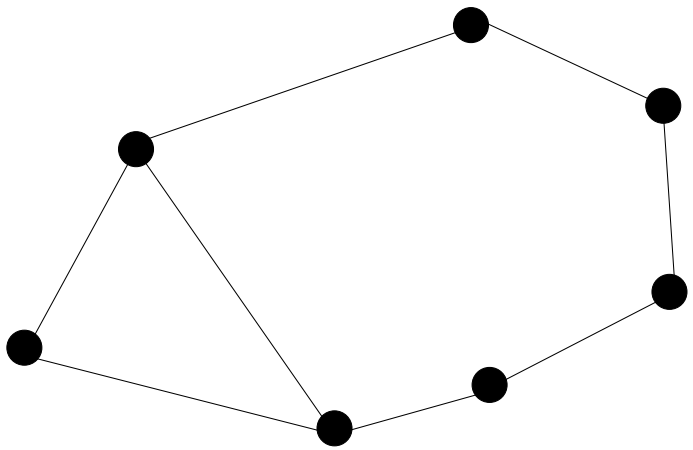
security of MANETs

- group communication in mobile ad hoc network using group key
- intrusion detection system (IDS) checks for compromised nodes
- IDS may not detect (false negative)
- IDS may erroneously detect (false positive)
- IDS may correctly detect



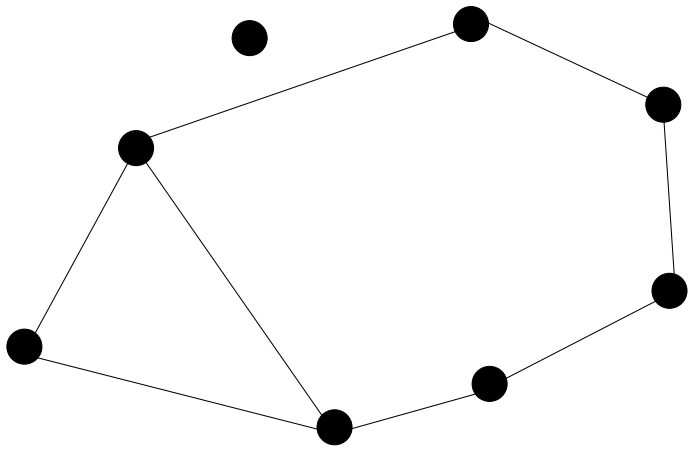
security of MANETs

- group communication in mobile ad hoc network using group key
- intrusion detection system (IDS) checks for compromised nodes
- IDS may not detect (false negative)
- IDS may erroneously detect (false positive)
- IDS may correctly detect and remove



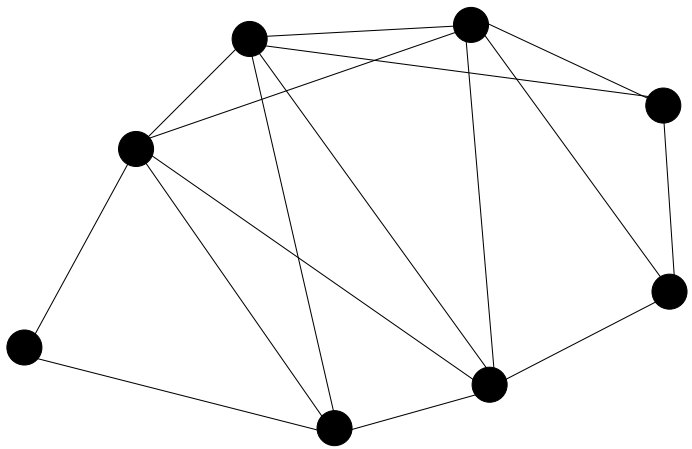
security of MANETs

- group communication in mobile ad hoc network using group key
- intrusion detection system (IDS) checks for compromised nodes
- IDS may not detect (false negative)
- IDS may erroneously detect (false positive)
- IDS may correctly detect and remove
- node is excluded



security of MANETs

- group communication in mobile ad hoc network using group key
- intrusion detection system (IDS) checks for compromised nodes
- IDS may erroneously detect (false positive)
- IDS may correctly detect and remove
- node is excluded
- new node arrives and is included
- key change is necessary to maintain secure communication



Performance analysis of dynamic group communication systems with intrusion detection integrated with batch rekeying in mobile ad hoc networks. J.-H. Cho, I.-R. Chen, and P.-G. Feng. AINAW '08: Proceedings of the 22nd International Conference on Advanced Information Networking and Applications – Workshops, pp. 644–649, Washington, DC, USA, 2008.

rekeying in MANETs

intrusion detection

- voting-based intrusion detection
- byzantine failure, more than 1/3 of nodes compromised

rekeying frequency

- rekeying increases security
- rekeying increases load (cost)
- batch rekeying after n membership changes

rekeying in MANETs

intrusion detection

- voting-based intrusion detection
- byzantine failure, more than 1/3 of nodes compromised

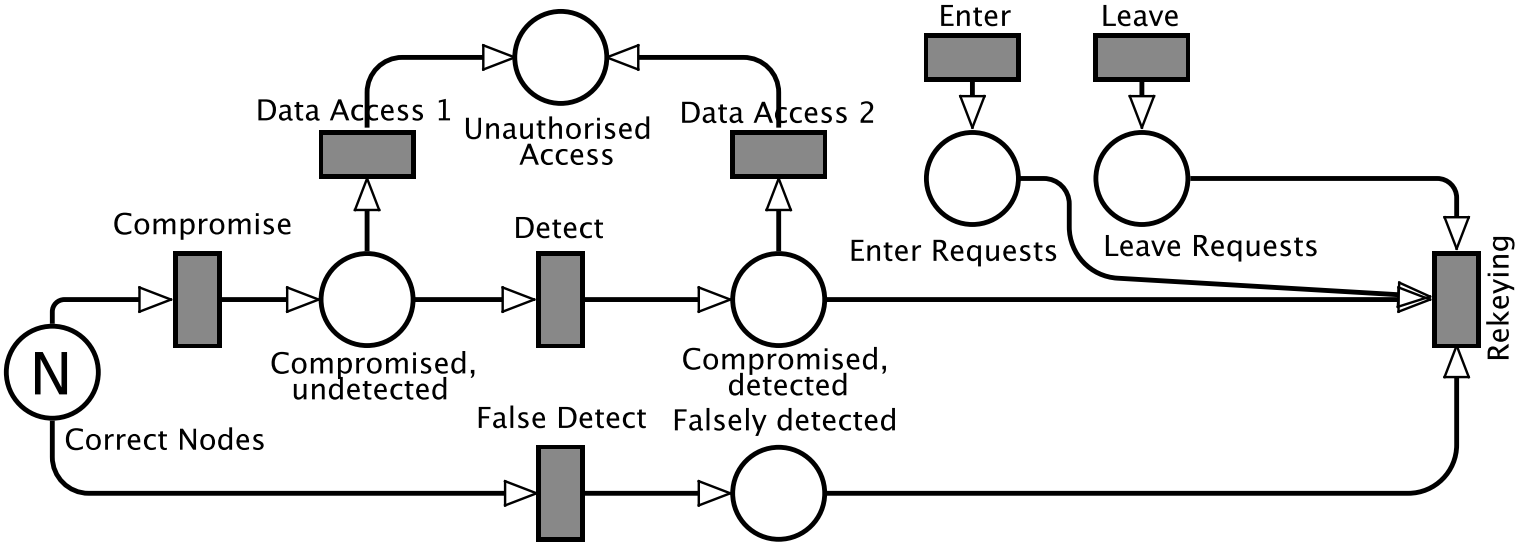
rekeying frequency

- rekeying increases security
- rekeying increases load (cost)
- batch rekeying after n membership changes

optimisation problem

how often to change key for optimal performance and security?

Petri net model



parameters

- $k1$ rekey limit on (trusted) join and leave requests
- $k2$ rekey limit on detected and falsely detected compromised nodes

measures

performance measure

average response time R of transmitted message

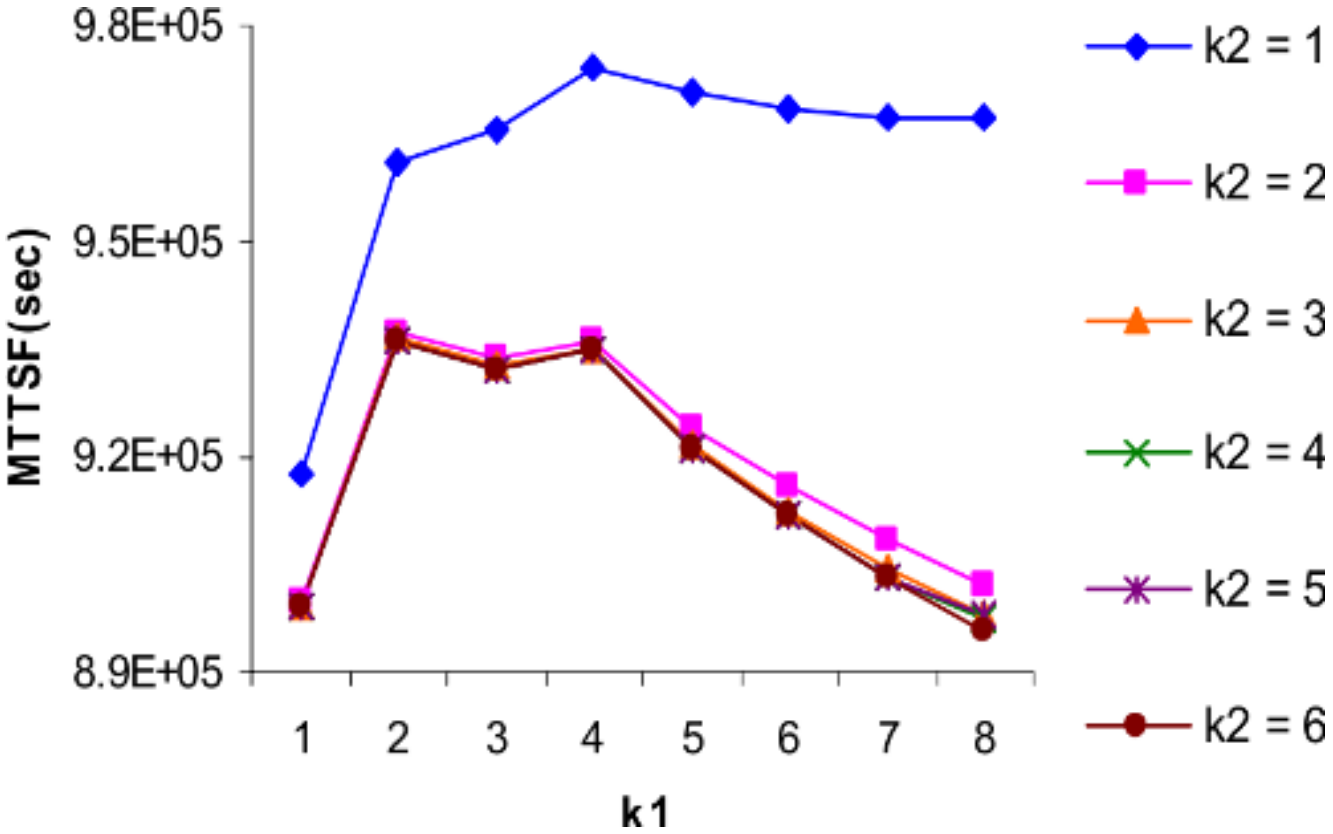
security measure

MTTSF (attacker takes over or system becomes unavailable, more than 1/3 compromised nodes)

computation method

- analysis of SPN
- MTTA method (mean time to absorption)

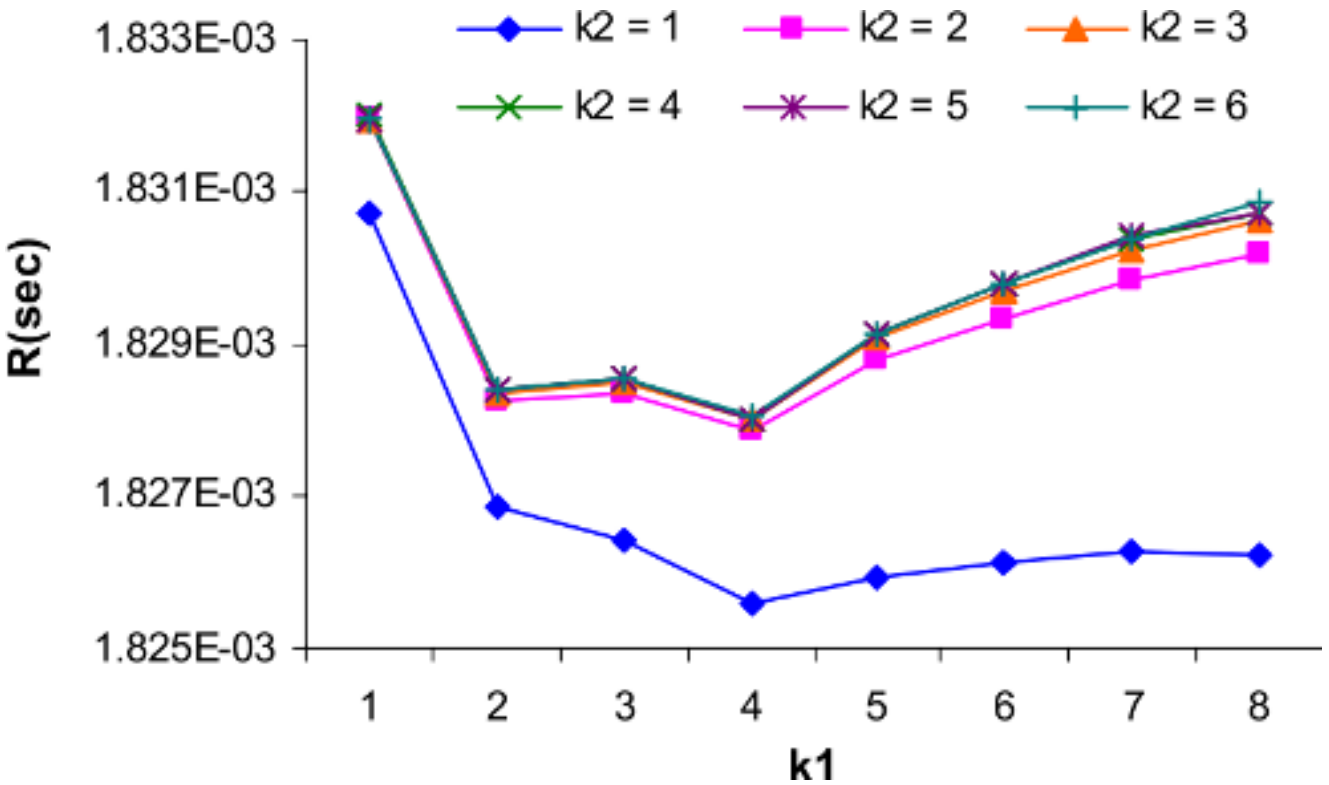
mean time to security failure



parameters

- $k1$ rekey limit on (trusted) join and leave requests
- $k2$ rekey limit on detected and falsely detected compromised nodes

response time



vary rekeying thresholds

- rekeying limit at 4 join/leave requests seems optimal
- for higher detected/falsely detected limit 2 join/leave requests might be better
- either consider less join/leave requests, or less detected/falsely detected nodes?

intrusion detection interval

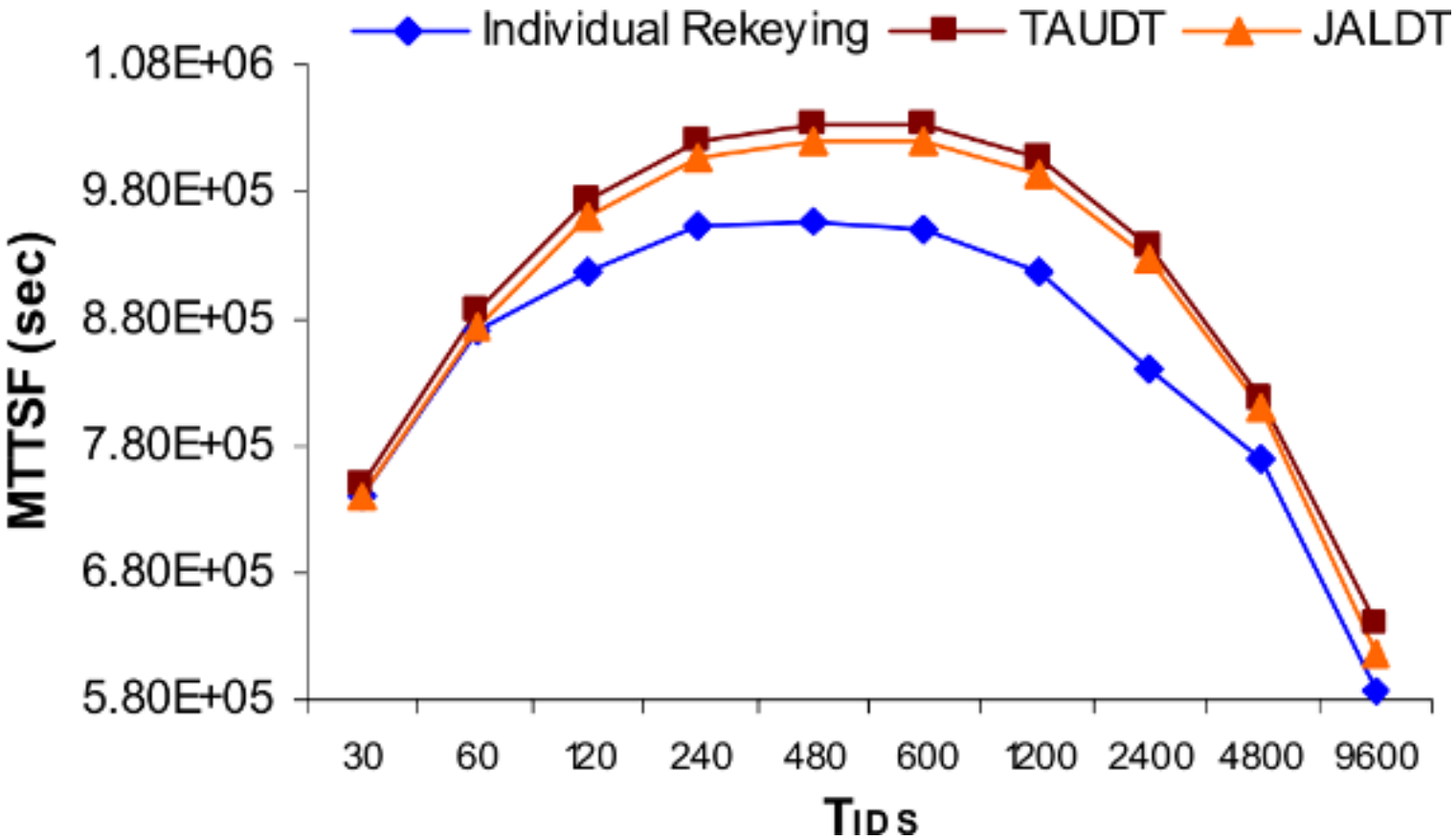
rekeying strategies

- individual rekeying (after each join, leave, evict event)
- threshold-based rekeying
  - TAUDT, $k1$, $k2$ as above
  - JALDT, $k1$ = limit on join requests, $k2$ = limit in leave requests and evicted nodes.

parameters

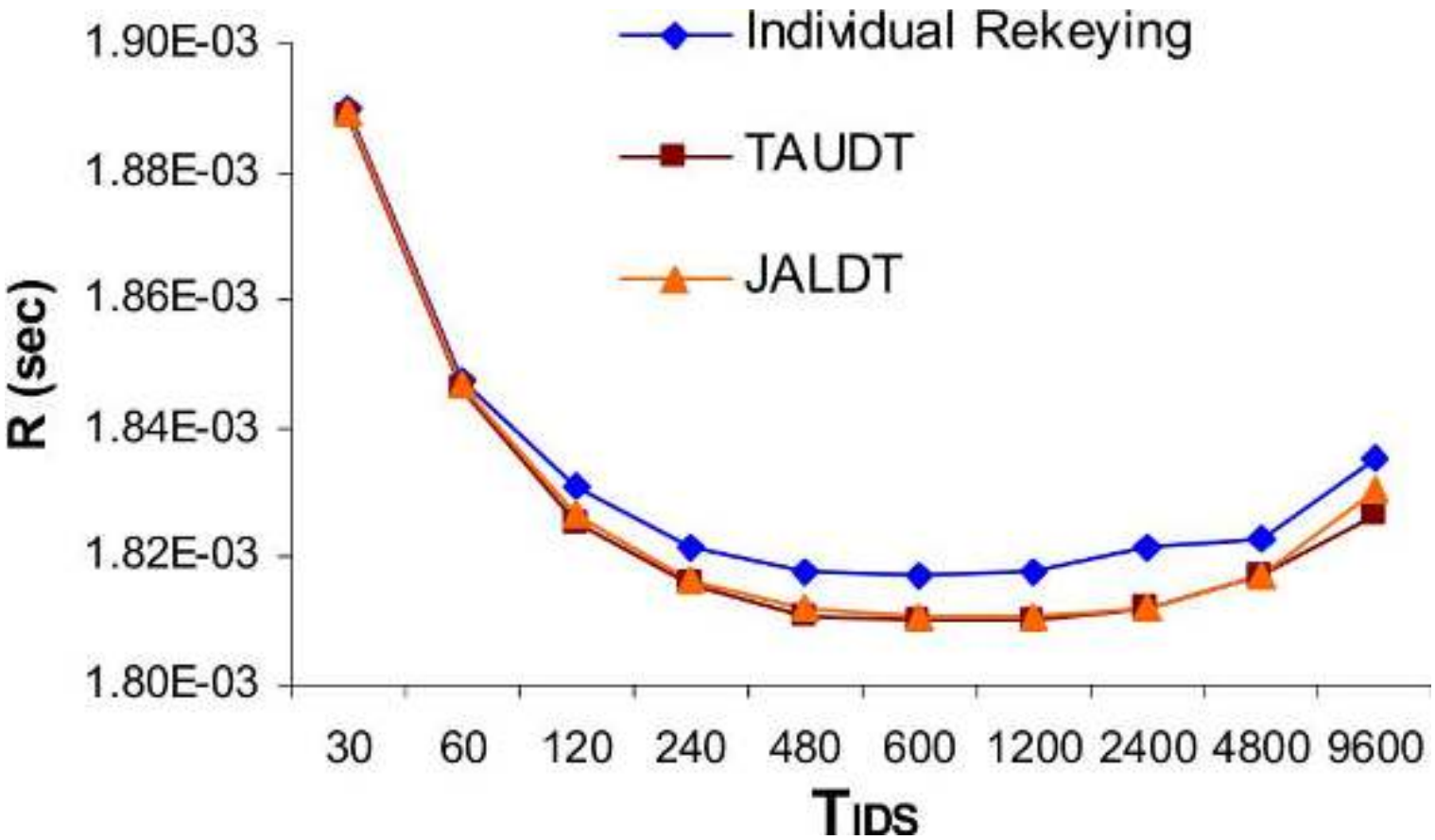
- investigate optimal IDS interval (firing time)
- set TAUDT: (k1, k2) = (4,1), JALDT: (k1, k2) = (5,2) (enabling condition)

optimal intrusion detection time



- $T_{IDS} = 480$ optimises MTTSF for individual rekeying
- $T_{IDS} = 600$ optimises MTTSF for threshold-based rekeying

optimal intrusion detection interval



- $T_{IDS} = 600$ optimises response time for all rekeying strategies

conclusions

results

- security and performance of wireless group communication system
- security is measured in terms of MTTSF
- performance is measured in terms of response time
- intrusion detection threshold and
- intrusion detection interval are chosen as to optimise those measures

Security of the email system

Security of the email system

considered system

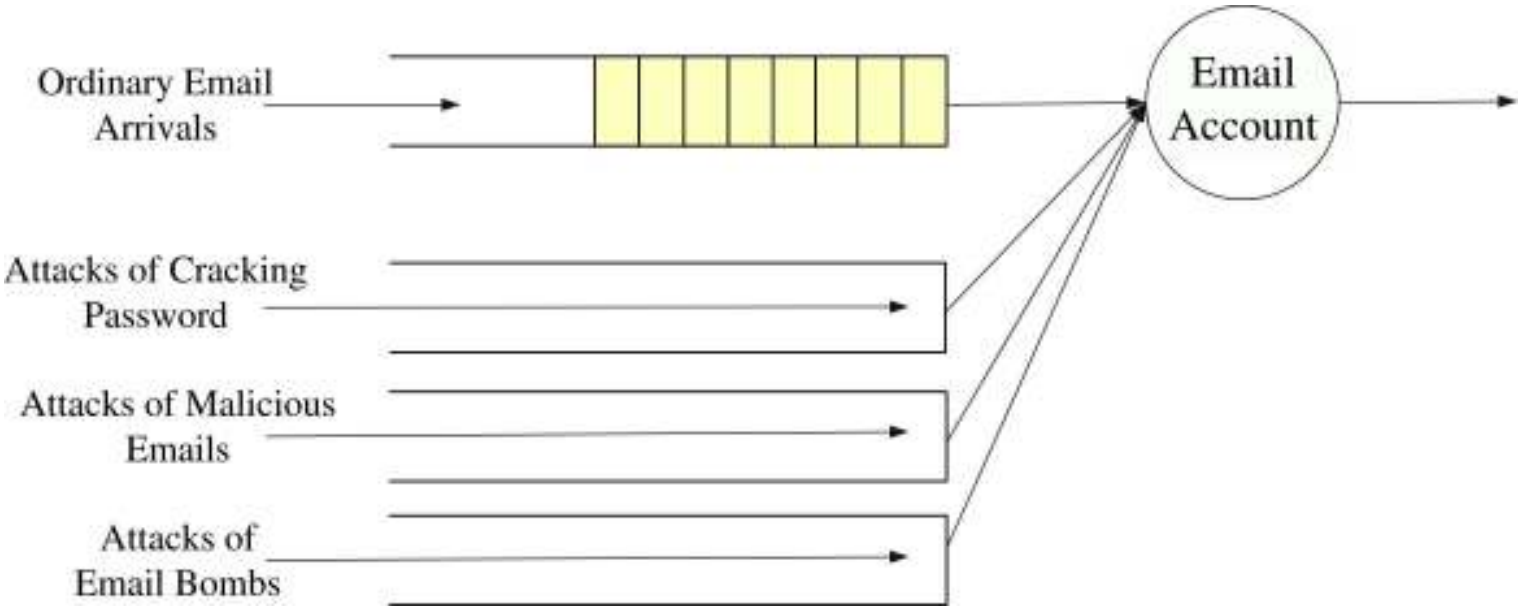
- email system considered a queue
- Inbox, filtering mechanisms, user, ...?

attack types

- gather information (malicious access to mailbox, click on link in malicious email)
- denial of service (email bombs flood the mail system)

Y. Wang, C. Lin, and Q.-L. Li. Performance Analysis of the Email System under Three Types of Attacks. Performance Evaluation, 67(6), (June 2010)

multiple queues

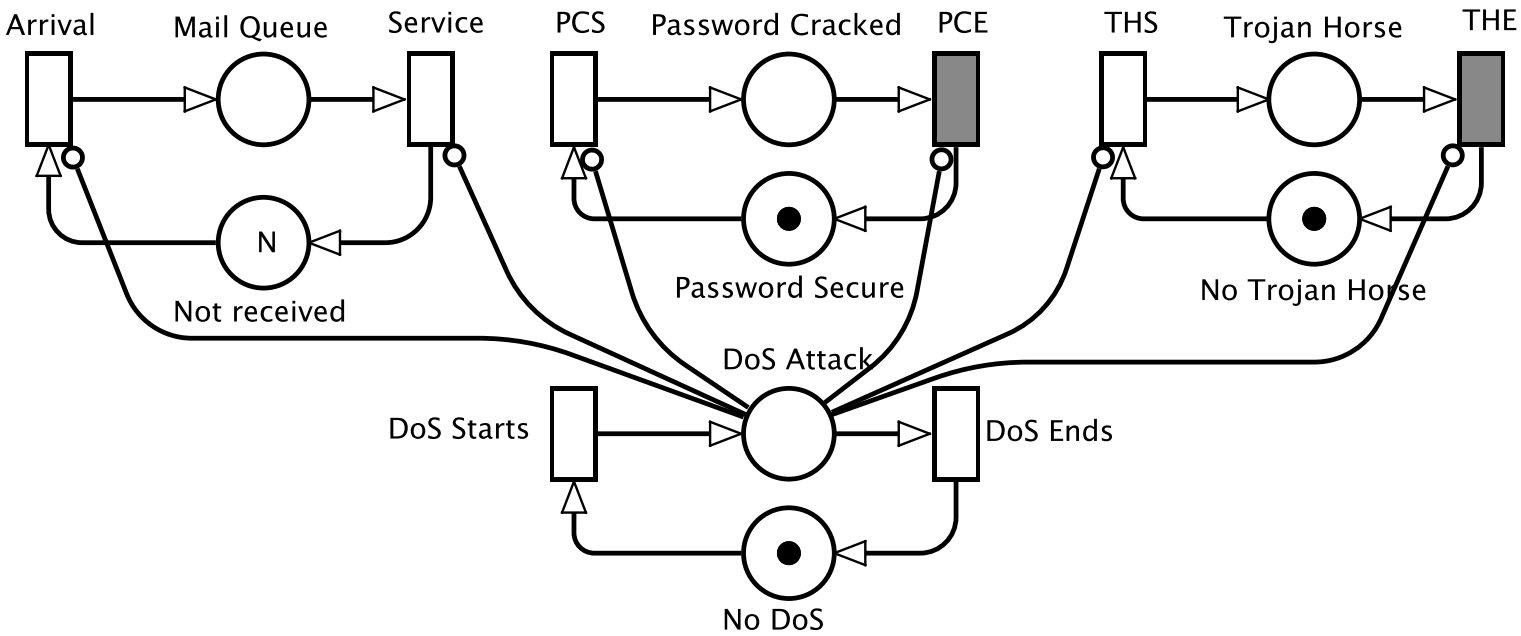


parameters

each queue is described by arrival and service time distribution/rate

- emails, M/M/1/N: λ, μ
- Cracking password, M/PH/1/1: α_c and (γ_c, S_c)
- Malicious email, M/PH/1/1: α_m and (γ_m, S_m)
- Email bombs, M/M/1/1: α_b, β_b

Petri net model



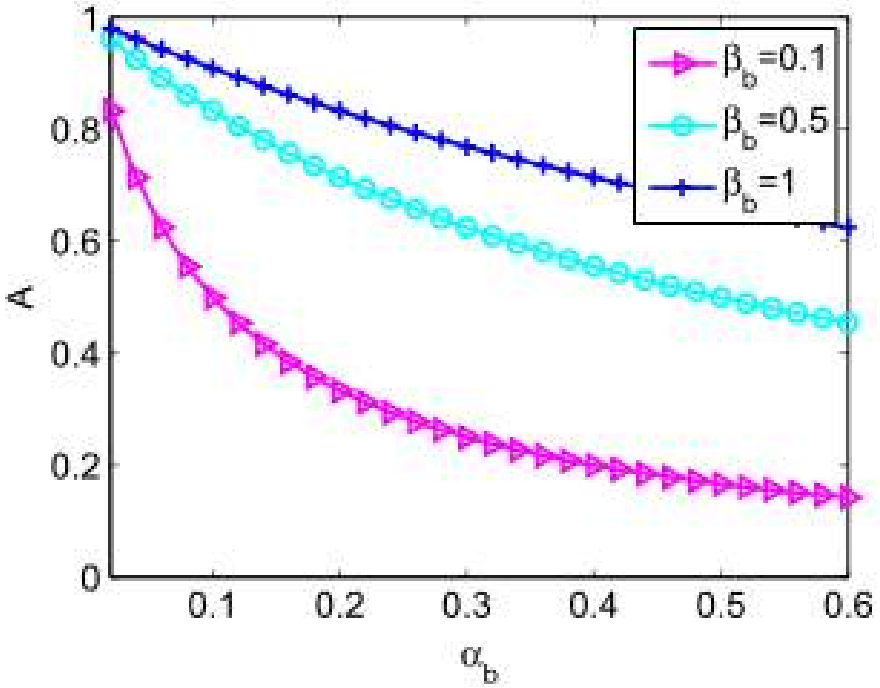
performance measure

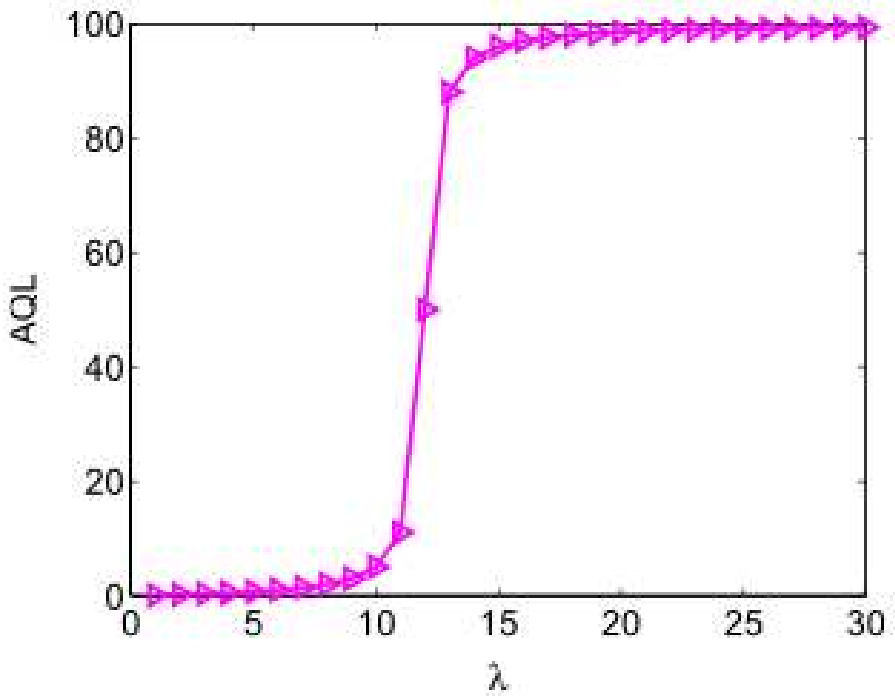
- queue length
- system availability

security measure

- (availability)
- information leakage probability

performance measures

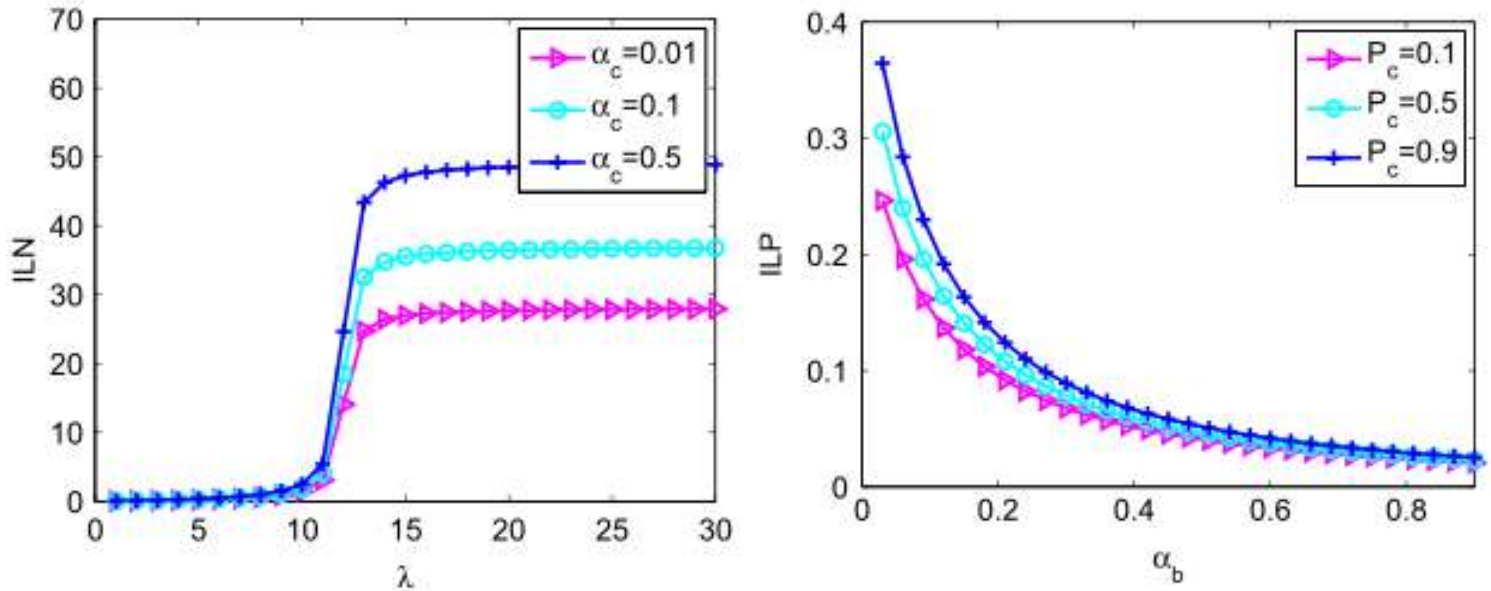




availability and queue length

- availability versus arrival rate of email bombs for different damage duration
- average queue length versus email arrival rate

security measures



information leakage

- information leakage versus email arrival rate for different arrival rates of cracking attacks
- information leakage probability versus email bomb arrival rate for different probabilities of obtaining information after cracking the password.

security of email

- malicious emails are known security concern
- formalisation as finite queueing models doubtful
- provided performance as well as security measures
- availability, queue length, information leakage

Modelling Performance Security Tradeoff

performance and security model

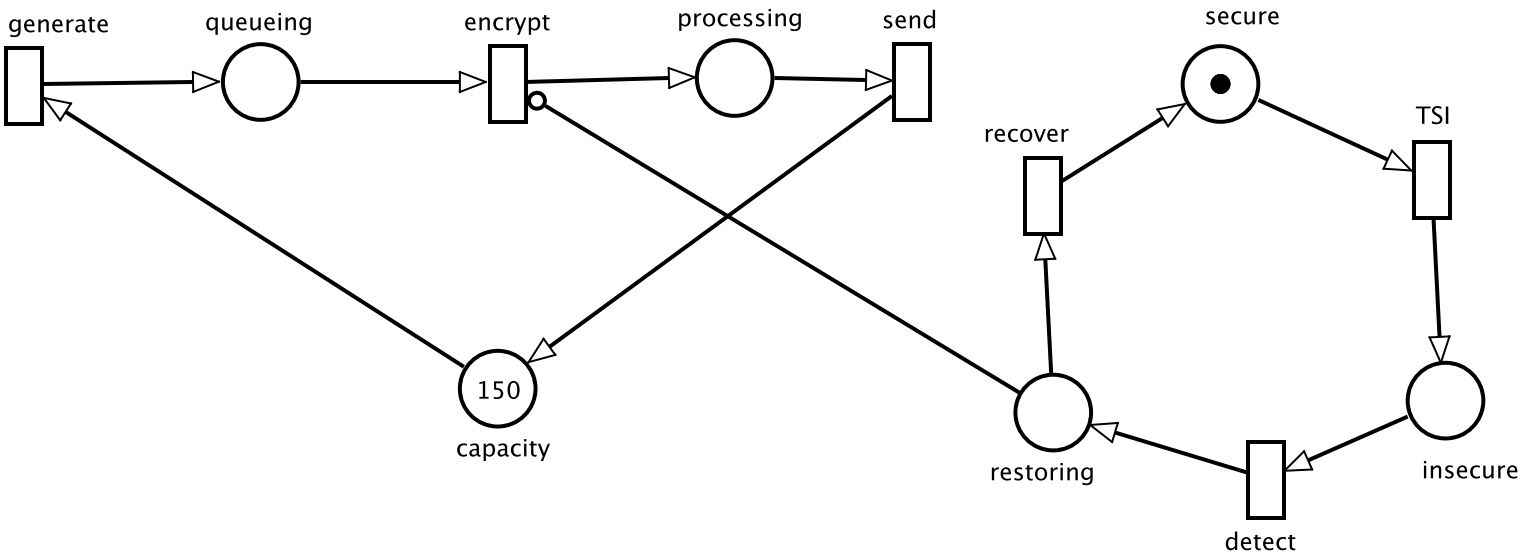
objective

- separate performance and security models
- combined measures with optima (cf. performability)
- example: encryption of messages (recall Lamprecht et al.)
- assumption: longer keys $\rightarrow$ more secure, longer encryption time

model specification

- performance model (queue)
- security model (CTMC, ...)

Petri net model



parameters

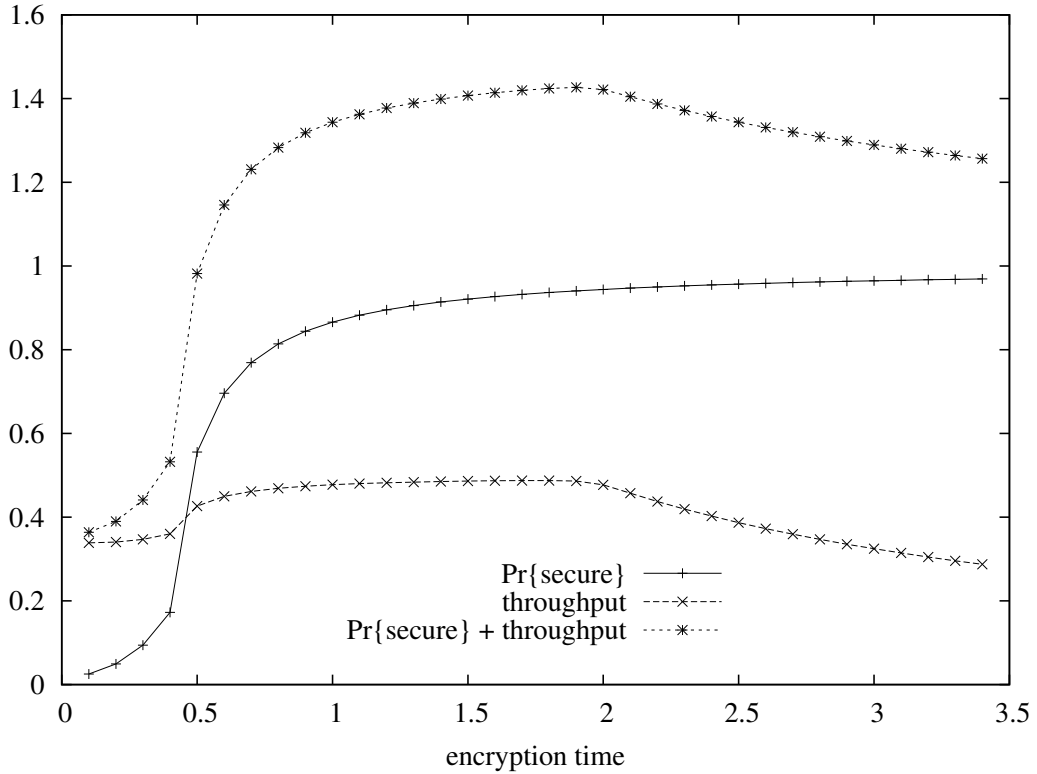
Parameter Name	Value/Delay
generate	2.0
send	0.1
N	150
encrypt	0.1, . . ., 3.4 by 0.1
TSI	12.5,25,50,100, . . ., 15100 by 500
detect	120
recover	360

combine performance and security

- pure performance measure (throughput)
- pure security measure (prob. secure state)
- combined measures involving costs

Throughput(*send*)	$10 \cdot \Pr\{\#processing > 0\}$
$\Pr\{secure\}$	$\mathrm{E}[\#secure] = \Pr\{\#secure > 0\}$
CPSM	Throughput(*send*) $+ \Pr\{secure\}$
Gain	$2 \cdot \mathrm{E}[\#processing \text{ IF } \#secure = 1]$
Loss	$-\mathrm{E}[\#processing \text{ IF } \#insecure = 1]$
lowCostRevenue	$2 \cdot \mathrm{E}[\#processing \text{ IF } \#secure = 1] - \mathrm{E}[\#processing \text{ IF } \#insecure = 1]$
highCostRevenue	$\mathrm{E}[\#processing] \cdot (2 \cdot \mathrm{E}[\#secure] - 5 \cdot \mathrm{E}[\#insecure])$

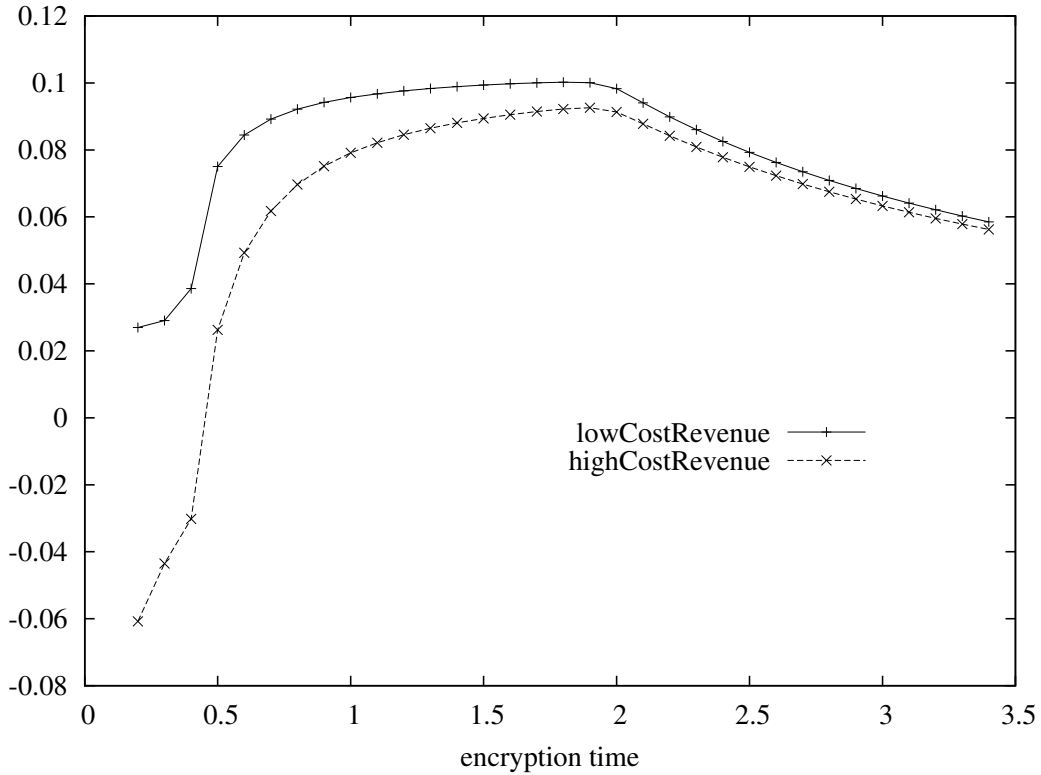
analysis



results

- Pr(secure) and throughput both *high better metrics* (Raj Jain)
- sum is HB as well

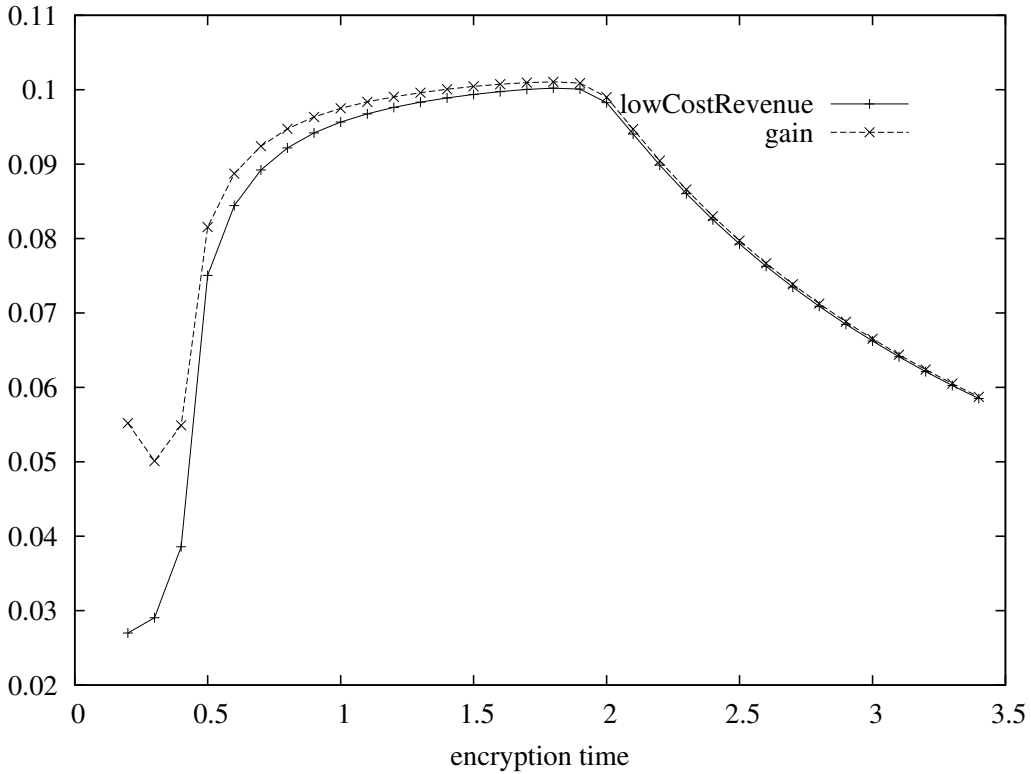
indirect measures



penalties

- higher penalty $\Rightarrow$ lower benefit
- optimum key length is the same

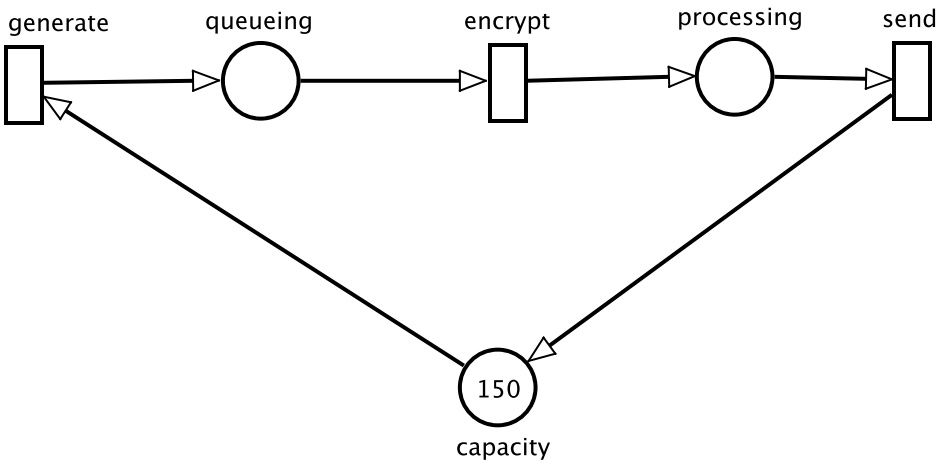
encryption cost

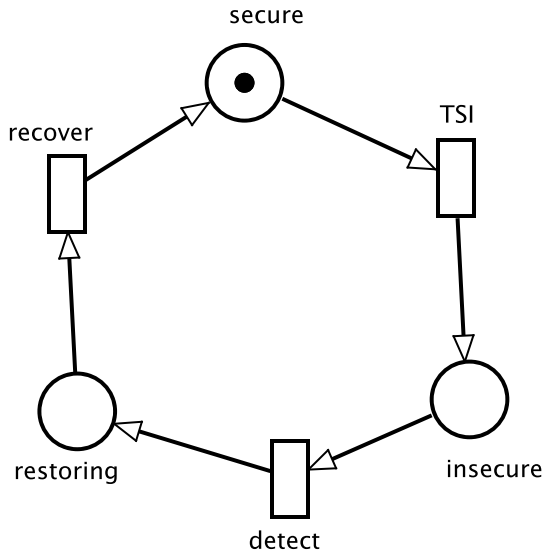


encryption cost

- cost = revenue - gain
- cost negligible for long keys
- cost of security failure

simplified Model

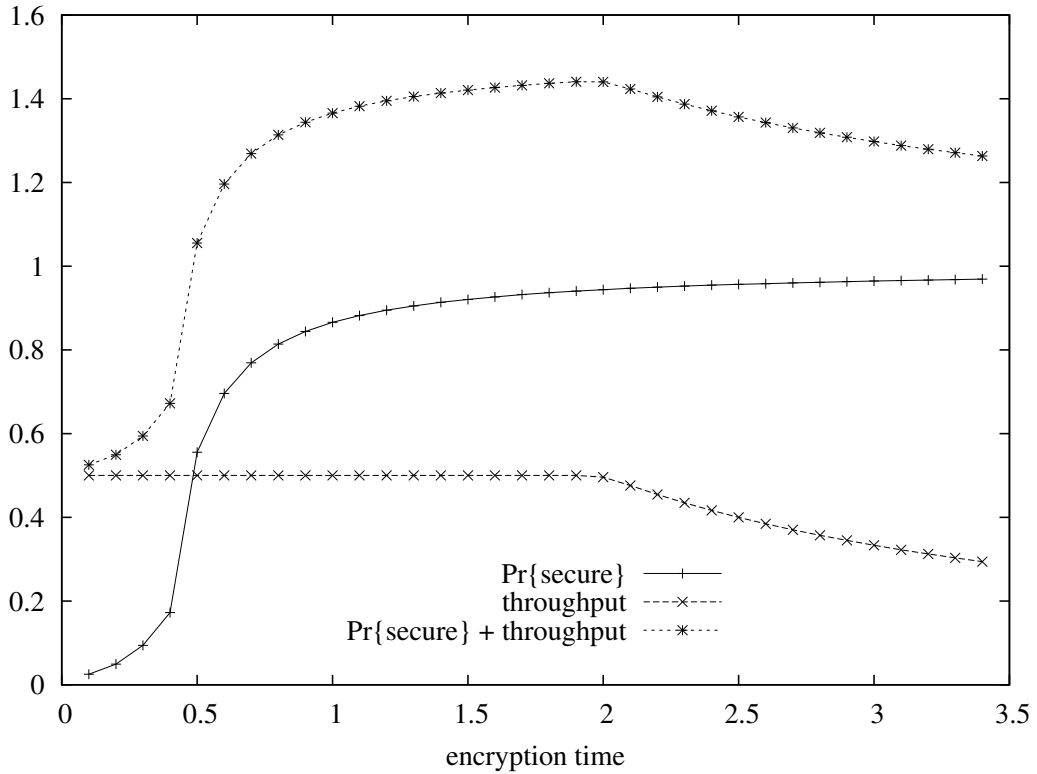




separation of performance and security model

- what happens if we keep the submodels completely separate?
- monotonous performance and security measures?

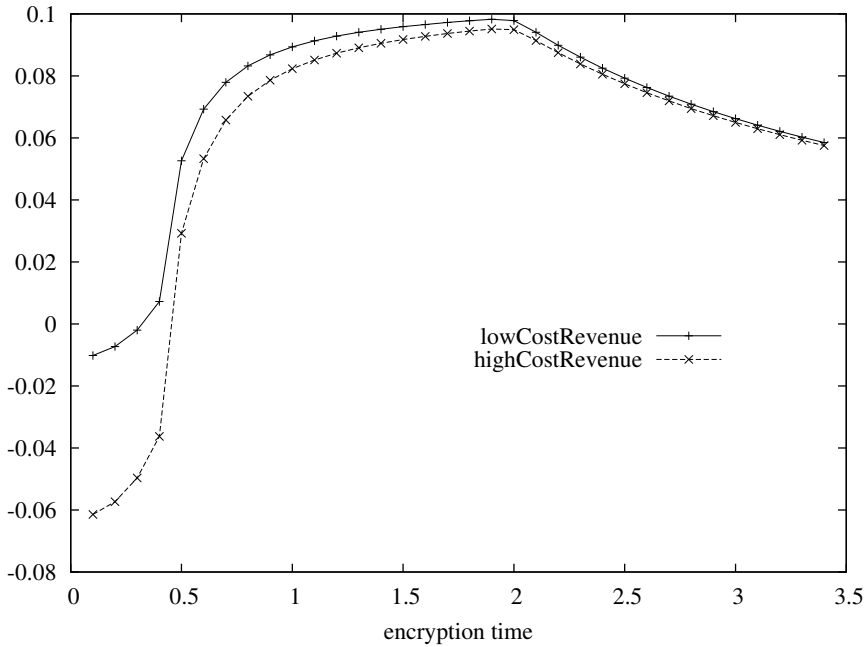
simplified model throughput

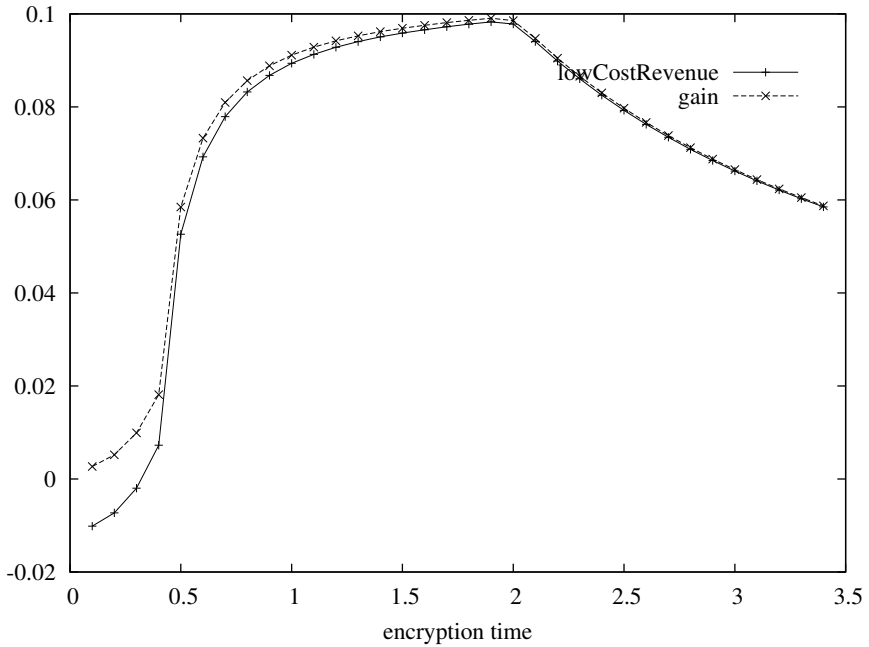


combined performance and security measure

- limiting arrival process more pronounced
- throughput unaffected

simplified model revenue





insights

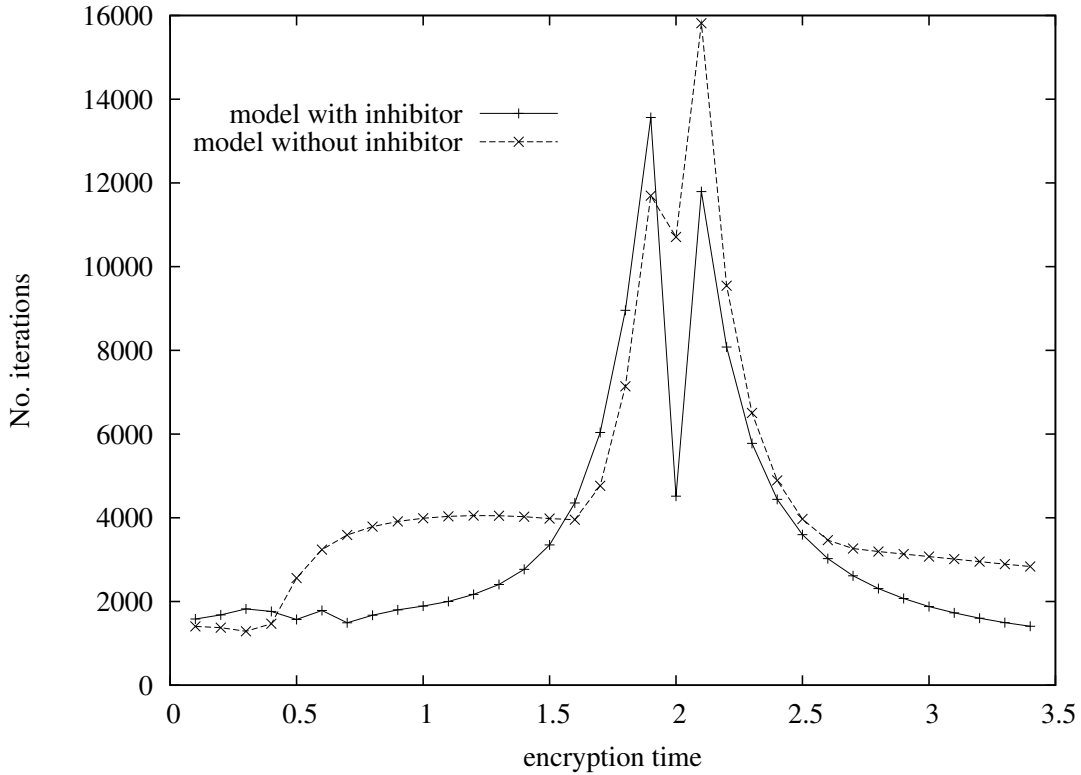
lessons learnt

- assumptions made: TSI and encryption time are correlated
- processing discontinues/continues in case of recovery, what about the measures?
- do we gain information beyond the assumptions made initially?

parameters

- we find optimal parameter settings!!
- how about realistic parameter values?

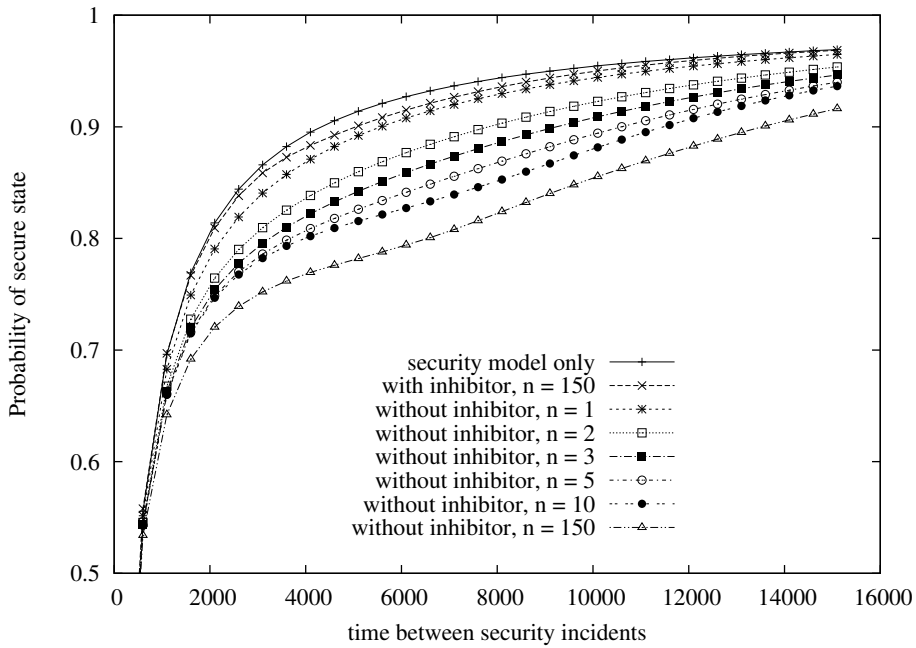
numerical issues

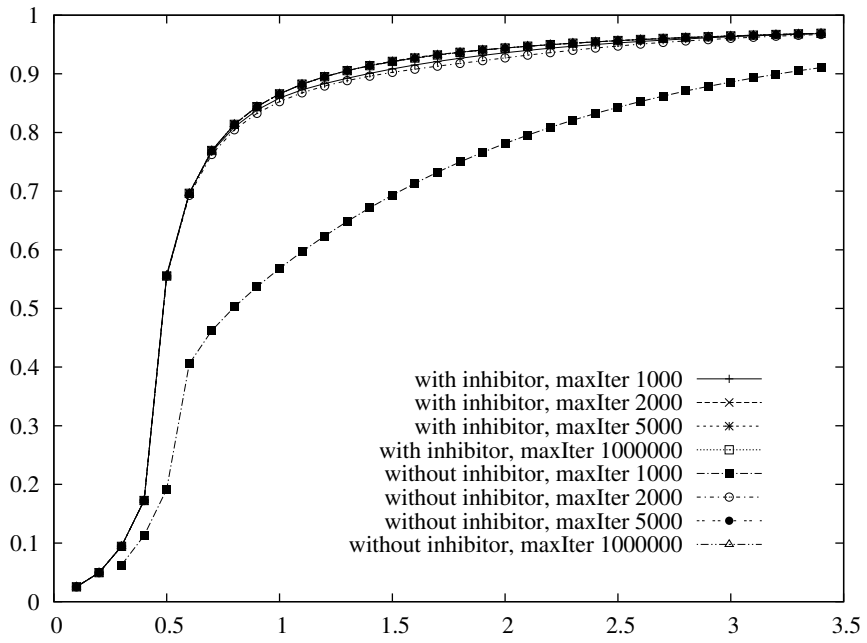


remember performability

- many iterations needed
- poor accuracy

numerical issues





- solution sensitive to queue length
- solution sensitive to no. of iterations

conclusions

quantify security

- model-based analysis of performance and security is a new field although the issue has been around for long
- we still have no metric for security, but
- frequent change of key, or ticket increases security
- longer keys for encryption increase security
- performance can be measured using throughput and response time
- tradeoff can be formulated

security statement

- cryptographic algorithms are known to be secure
- security problems are dependability problems (overflow, implementation, failures, etc.)

model results

- do we find out something about the system, or about the model?
- setting up a good model is very difficult.

model results

- do we find out something about the system, or about the model?
- setting up a good model is very difficult.

resume

do we lie with stochastics?

model results

- do we find out something about the system, or about the model?
- setting up a good model is very difficult.

resume

do we lie with stochastics?