# Energy Aware Real-Time Scheduling Policy with Guaranteed Security Protection

Wei Jiang[1], Ke Jiang[2], Xia Zhang[1], Yue Ma[3]

[1]School of Information and Software Engineering
University of Electronic Science and Technology of China
[2]Department of Computer and Information Science
Linköping University, Sweden
[3]Department of Computer Science and Engineering
University of Notre Dame, USA

# Introduction

- **Embedded system design concerns**

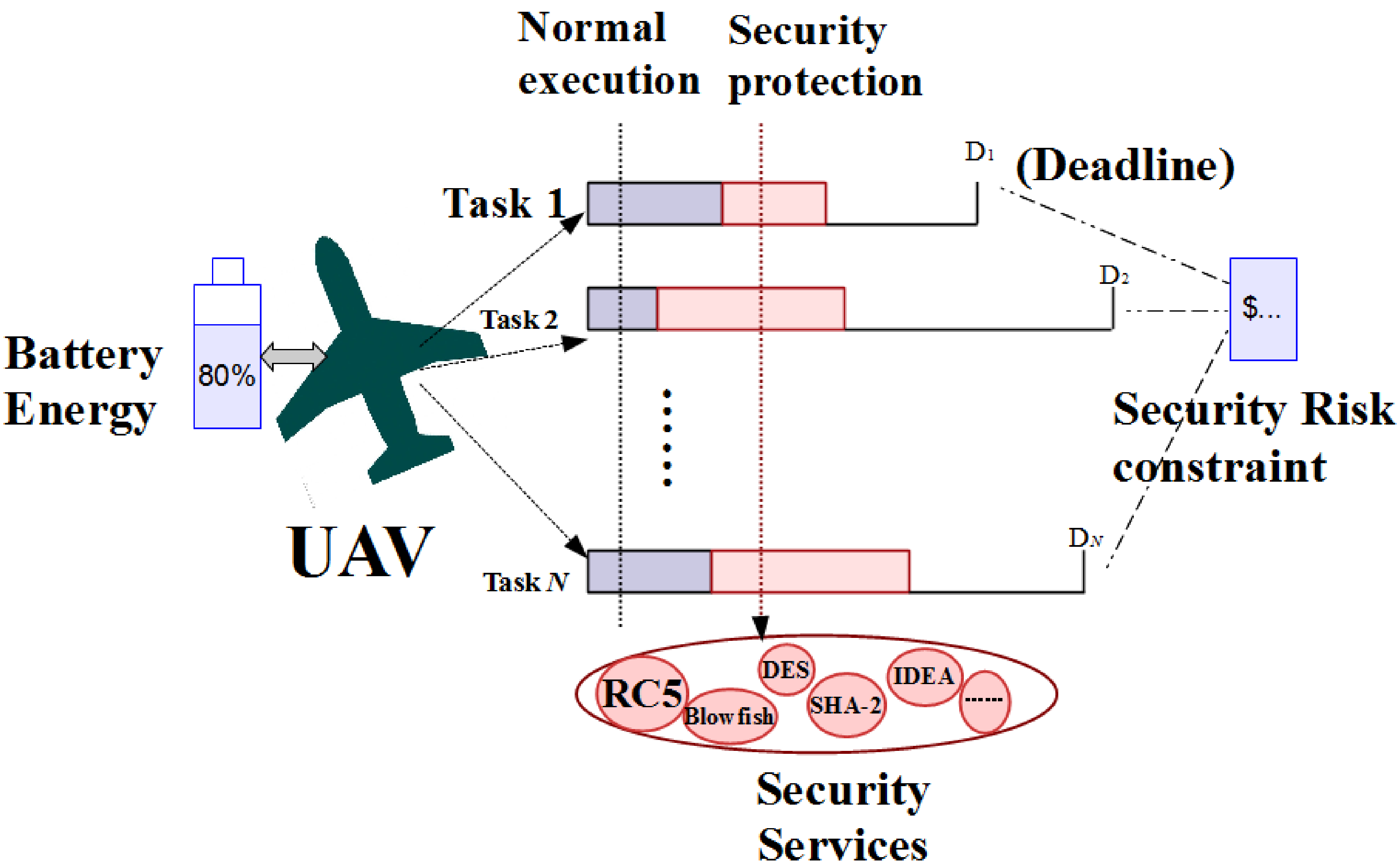
# Contributions

- **The Design Problem**
  - Security- & Energy-aware Real-Time Application
  - System execution goal
    - Complete the App. with minimal energy
    - Satisfy the security and real-time requirements
  - NP-hard to find the best solution
- **The Method**
  - Dynamic Programming based Approximate optimization framework

# Outline

- Motivational application
- System model
- Problem formulation
- Approximation based Dynamic Programming
- Experimental results
- Conclusion

# Motivational application



# System Model

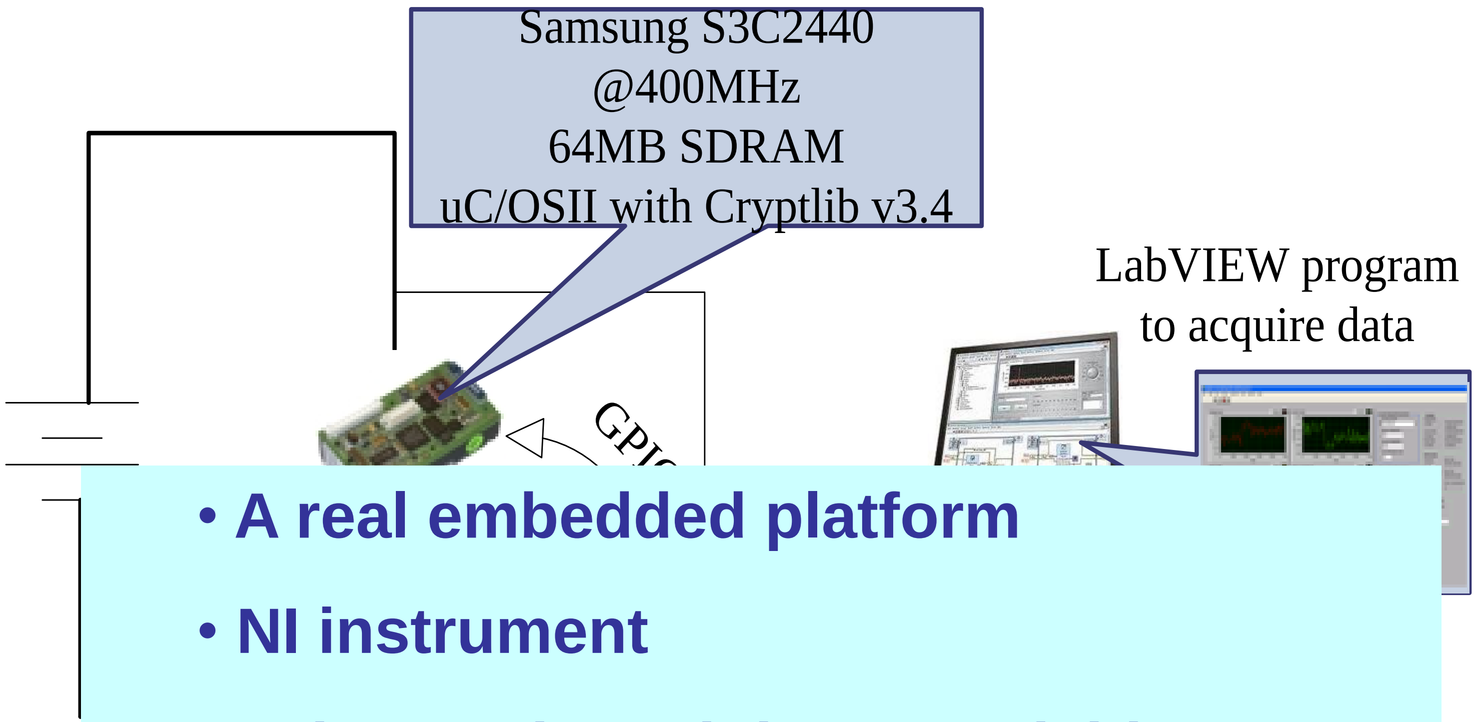
- Architecture model
  - Mono-processor, Battery powered

- Application model
  - A set of periodic security- & energy-aware tasks
  - Security risk constraint
  - Scheduling by classic method, e.g. RM/EDF

- Task model
  - A mandatory and optional part (Security improve)
  - Task attributes: $(BE_i, P_i, L_i, S_i, S_i^{DM}, V_i, SR_i)$

# Security Overhead Model

- Measure energy & time of  security algorithms




- A real embedded platform
- NI instrument
- LabVIEW based data acquisition
- Nearly non-intrusive measurement

# Security Overhead Model

- **Measurement results**

| Ciphers | time(ms/KB) | Energy(mJ/KB) | Sec. Level |
| --- | --- | --- | --- |
| RC4 | 0.0063 | [illegible] | [illegible] |
| RC5 | 0.0125 | [illegible] | [illegible] |
| BLOWFISH | 0.0170 | [illegible] | [illegible] |
| IDEA | 0.0196 | [illegible] | [illegible] |
| SKIPJACK | 0.0217 | [illegible] | [illegible] |
| 3DES | 0.0654 | 21.0914 | 6 |

**Energy/time ratio: 320 mJ/S**

**POW (power)**

- Execution time of each task
- $Exe_i = BE_i + \theta(S_i) * L_i$
- Energy consumption of each task
- $En_i = POW * (BE_i + \theta(S_i) * L_i)$

# Security Risk Model of each task

- **Definition**
  - **Security risk (SR)** is the product of security failure probability and consequence impact of security failure.
  - $$SR_i = Pro_i^{risk} * V_i$$
  - **Failure probability**
  - $$Pro_i^{risk} = \begin{cases} 0, & if\ S_i \geq S_i^{DM} \\ 1 - e^{-\lambda_i (S_i^{DM} - S_i)} \end{cases}$$
  - More reasonable than other linear security QoS definitions like ref. [8, 10]

# Problem formulation

- **Original problem**

**Min** $\text{Energy} = \sum_{i=1}^{N}\left(\frac{HP}{P_i}\right) * En_i$

**S.T.** $\begin{cases} \sum_{i=1}^{N}\left(\frac{HP}{P_i}\right) * SR_i \leq RB \\ \sum_{i=1}^{N}(BE_i + \theta(S_i) * L_i)/P_i \leq UB_x \\ S^{min} \leq S_i \leq S^{max} \end{cases}$

- $\text{Energy} = \sum_{i=1}^{N}\left(\frac{HP}{P_i}\right) * En_i$

$$= HP * POW * \sum_{i=1}^{N}(BE_i + \theta(S_i) * L_i)/P_i$$

# Problem formulation

- **Reduced problem**

$$\textbf{Min} \quad \sum_{i=1}^{N} (BE_i + \theta(S_i) * L_i)/P_i$$

$$\textbf{S.T.} \quad \begin{cases} \sum_{i=1}^{N} \left(\frac{HP}{P_i}\right) * SR_i \le RB \\ \sum_{i=1}^{N} (BE_i + \theta(S_i) * L_i)/P_i \le UB_x \\ S^{min} \le S_i \le S^{max} \end{cases}$$

  - **Min. Utilization**
  - **Risk constraint**
  - **Utilization cons**
  - **Security level c**

We don't need to consider **Energy** dimension!

# Proposed Optimization Technique

- Markov dynamic programming procedure
- Approximating policies and analysis
- Round Nearest approximating algorithm
- Low Time complexity

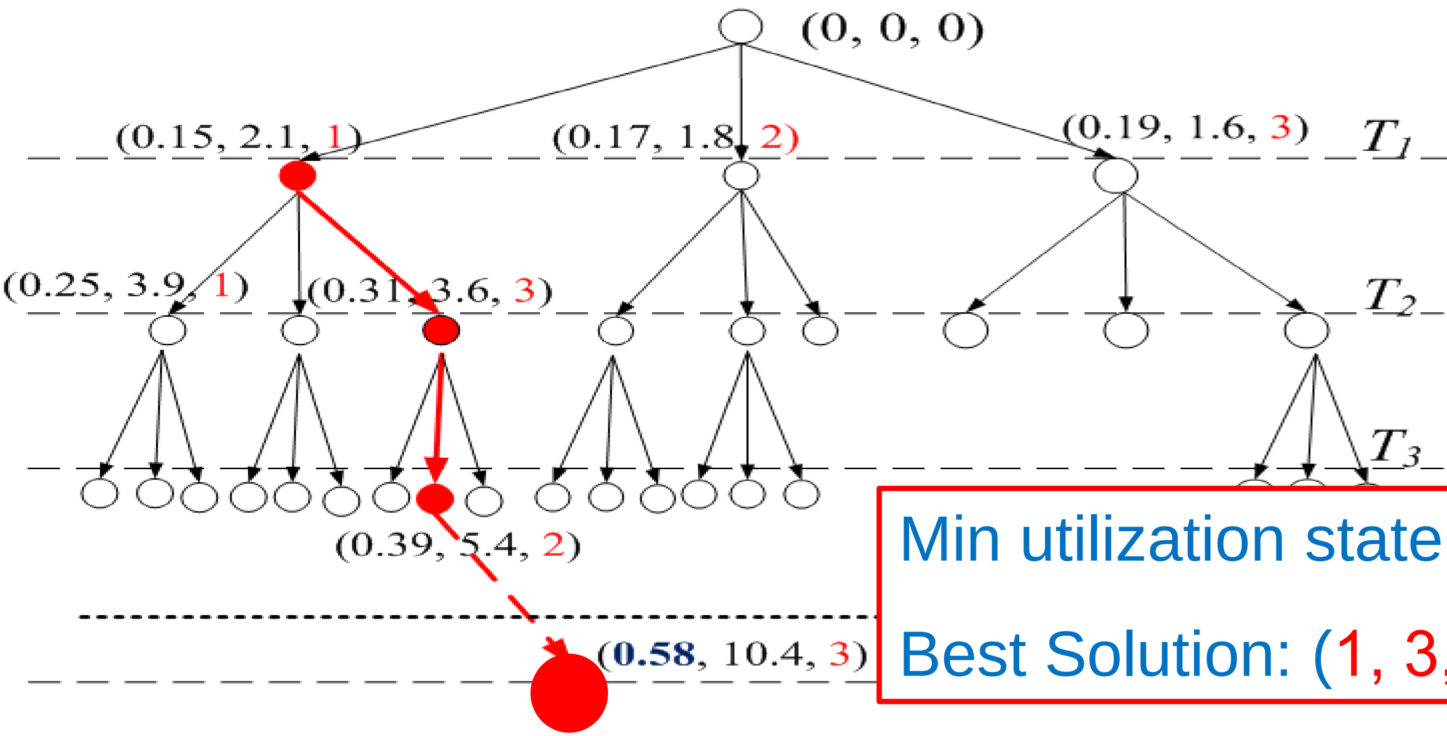
# Markov decision-making procedure

- Multi-stage decision procedure
  - N-Stage (One task, one stage)
  - Decision variable: $S_i$ (See level)
  - State definition: $(\xi_{ik}, \gamma_{ik}, S_{ik})$

Accumulated utilization ratio of first *i* tasks

Accumulated risk of first *i* tasks

Specific *level* for *k*-th state




Min utilization state

Best Solution: (1, 3, 2, …, 3)

# Multi-stage decision-making procedure

- Number of states increases exponentially! How next?
- Approximation of Knapsack problem
  - Scale risk into a series of discrete integers by $\Delta$
  - Replace states with same risk by lowest utilization one
  - States denoted by a $N \times M$ matrix, $M = \lceil RB/\Delta \rceil$

| 0 | 1 | 2 | … | $M$ |
|---|---|---|---|---|
| $T_1$ | $(\xi_{1,1}, \gamma_{1,1}, S_{1,1})$ | $(\xi_{1,2}, \gamma_{1,2}, S_{1,2})$ | … | $(\xi_{1,M}, \gamma_{1,M}, S_{1,M})$ |
| $T_2$ | $(\xi_{2,1}, \gamma_{2,1}, S_{2,1})$ | $(\xi_{2,2}, \gamma_{2,2}, S_{2,2})$ | … | $(\xi_{2,M}, \gamma_{2,M}, S_{2,M})$ |
| … | … | … | … | … |
| $T_{N\text{-}1}$ | $(\xi_{N-1,1}, \gamma_{N-1,1}, S_{N-1,1})$ | $(\xi_{N-1,2}, \gamma_{N-1,2}, S_{N-1,2})$ | … | $(\xi_{N-1,M}, \gamma_{N-1,M}, S_{N-1,M})$ |
| $T_N$ | $(\xi_{N,1}, \gamma_{N,1}, S_{N,1})$ | $(\xi_{N,2}, \gamma_{N,2}, S_{N,2})$ | … | $(\xi_{N,M}, \gamma_{N,M}, S_{N,M})$ |

# Approximating policies and analysis

- Round to Ceiling (RC)

$$RC(SR_i) = \left\lceil \frac{SR_i}{\Delta} \right\rceil$$

RC(2.2) = 3

Err = 0.8

- Round to Floor (RF)

$$RF(SR_i) = \left\lfloor \frac{SR_i}{\Delta} \right\rfloor$$

RF(2.8) = 2

Err = 0.8

- Round Randomly (RR)

$$RR(SR_i) = \begin{cases} \left\lceil \frac{SR_i}{\Delta} \right\rceil \text{with probability } \rho_1 = \frac{SR_i}{\Delta} - \left\lfloor \frac{SR_i}{\Delta} \right\rfloor \\ \left\lfloor \frac{SR_i}{\Delta} \right\rfloor \text{with probability } \rho_2 = \left\lceil \frac{SR_i}{\Delta} \right\rceil - \frac{SR_i}{\Delta} \end{cases}$$

- Round to Nearest (RN)

$$RN(SR_i) = SR_i^{\Delta} = \begin{cases} \left\lceil \frac{SR_i}{\Delta} \right\rceil, \text{ if } \frac{SR_i}{\Delta} - \left\lfloor \frac{SR_i}{\Delta} \right\rfloor \geq 0.5 \\ \left\lfloor \frac{SR_i}{\Delta} \right\rfloor, \text{ if } \frac{SR_i}{\Delta} - \left\lfloor \frac{SR_i}{\Delta} \right\rfloor < 0.5 \end{cases}$$

RN(2.2) = 2

RN(2.8) = 3

# Approximating policies and analysis

- How to determine $\Delta$, given $(1+\beta) * RB$ approximation?
- Overall Deviation for *N* tasks is:
  - $OD^{RC} \geq -N\Delta$
  - $OD^{RF} \leq N\Delta$
  - $-N\Delta \leq OD^{RR} \leq N\Delta$
  - $-N\Delta/2 \leq OD^{RN} \leq N\Delta/2$
- For $(1+\beta)$ approximation, $|OD| \leq \beta \cdot RB$

• Max $\Delta$ of RN is twice larger than RF, RC and RR!

• Reduce the number of states by a half in decision-making procedure !

# Round Nearest approximating algorithm

**Algorithm 1** RN-based approximation algorithm

Step 1: Schedulability test
**if** $\sum_{i=1}^{N}(HP/P_i)SR_i(S^{max}) > RB$ or $\sum_{i=1}^{N} Exe_i(S^{min})/P_i > UB_x$ **then**
    Return. */*Given task set is not schedulable*/*

Step 2: Initialization
Compute the grouping factor $\Delta = 2\beta RB/N$ and $M = \lceil RB/\Delta \rceil$
Initialize state matrix $\Omega_{N\times M}$ with each element $\Omega_{i,j} = (0,0,0)$
Initialize $\Omega_1$ by calculate $(\xi_1, \gamma_1, S_1)$ with each $S_1 \in [S^{min}, S^{max}]$

Step 3: Update the state matrix in $N$-Stage decision procedure
**for** $i = 2$ to $N$ **do**
    **while** $(\xi_{i-1}, \gamma_{i-1}, S_{i-1}) \neq (0,0,0)$ in $\Omega_{i-1}$ **do**
        **for** $S_i' = S^{min}$ to $S^{max}$ **do**
            Calculate temporary state $(\xi_i', \gamma_i', S_i')$
            **if** $\xi_i' > UB_x$ or $\gamma_i' > RB$ **then**
                Ignore this state and break */*Schedulability or security violated*/*
            **if** state $\Omega_{i,j}, (j = \gamma_i')$ is not existed **then**
                $\Omega_{i,j} = (\xi_i', \gamma_i', S_i')$ */*Store new state*/*
            **else if** $\xi_i' < \xi_i$ in $\Omega_{i,j}$ **then**
                $\Omega_{i,j} = (\xi_i', \gamma_i', S_i')$ */*Keep state with smaller utilization*/*

Step 4: Find the minimal energy consumption solution
Find $\Omega_{N,j}^{*}$ with minimal utilization ratio $\xi_N^{*}$
Obtain the final security assignment decision $(S_1, S_2, \cdots, S_N)$ by backtracking
$Energy^{*} = \xi_N^{*} * HP * POW$ */*The minimal energy*/*

# Experimental results

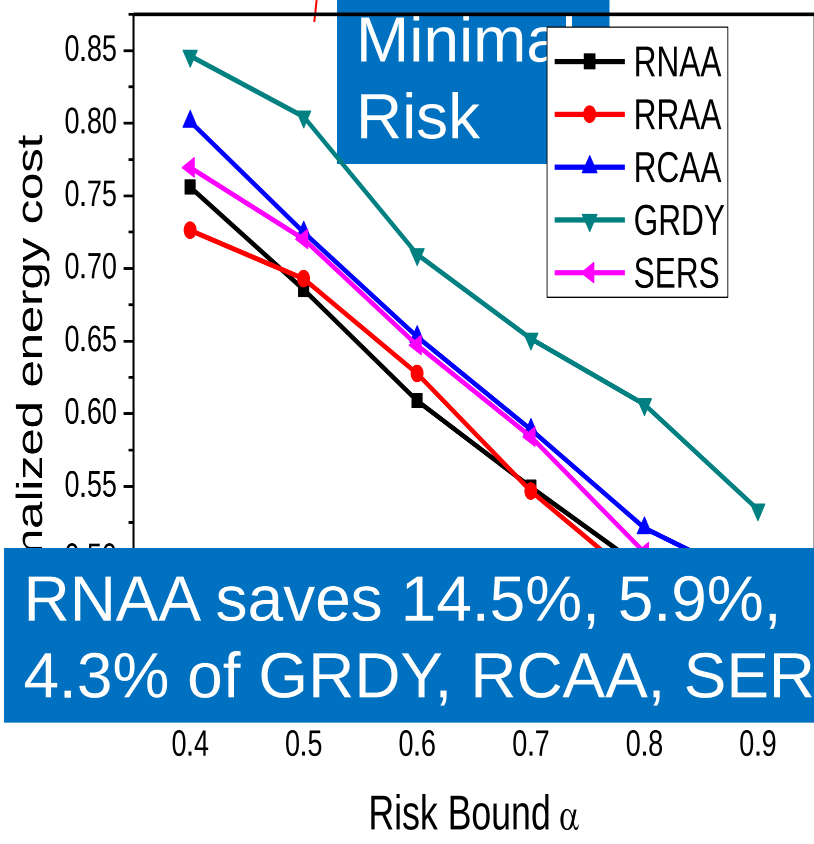
- **Experiment setup**
  - Two group simulations, each with three synthetic sets
  - Basic execution time of each task: 5~10 ms
  - Period: 300~500 ms
  - Confidential data size: 100~400 KB
  - Security demand: 6~8
  - Security impact/loss of each task: 5~10 $
  - Security coefficient $\lambda$ : 1~3

- **Compared algorithms**
  - RRAA: Round Randomly approximating algorithm
  - RCAA [16]: Round to Ceiling approximating algorithm
  - GRDY: Assigned security level in greedy fashion
  - SEAS [8]: Gradually increase the security by small risk/energy ratio

# Impacts of Risk Bound (RB)

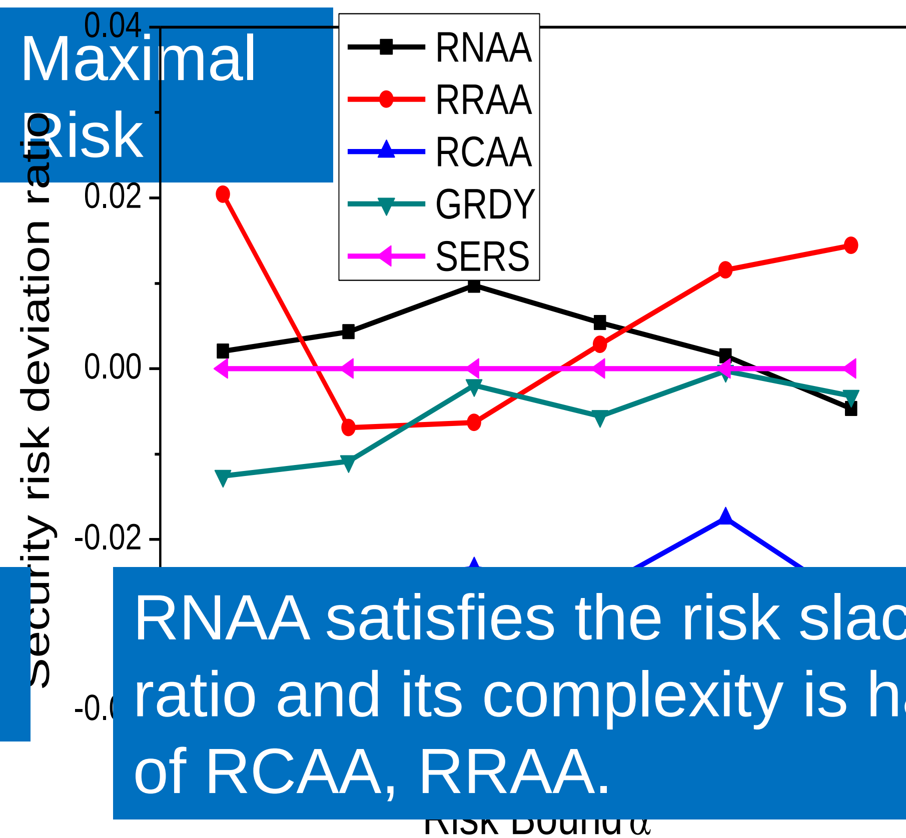
- $RB = MIR + \alpha * (MAR - MIR),\ \beta = 0.05$

Minimal Risk




RNAA saves 14.5%, 5.9%, 4.3% of GRDY, RCAA, SERS

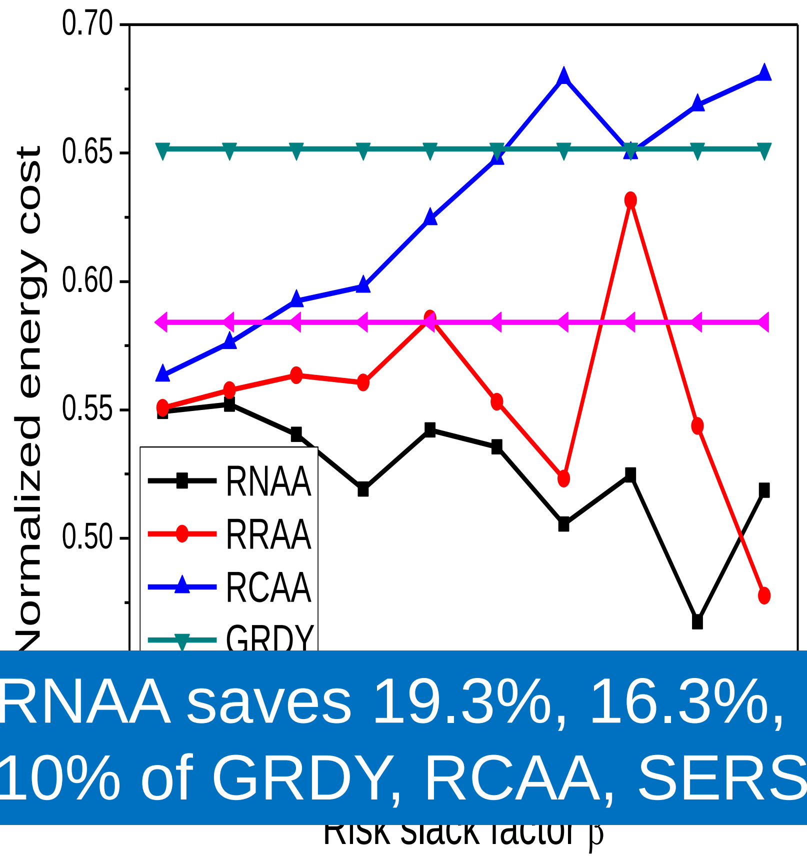
Maximal Risk



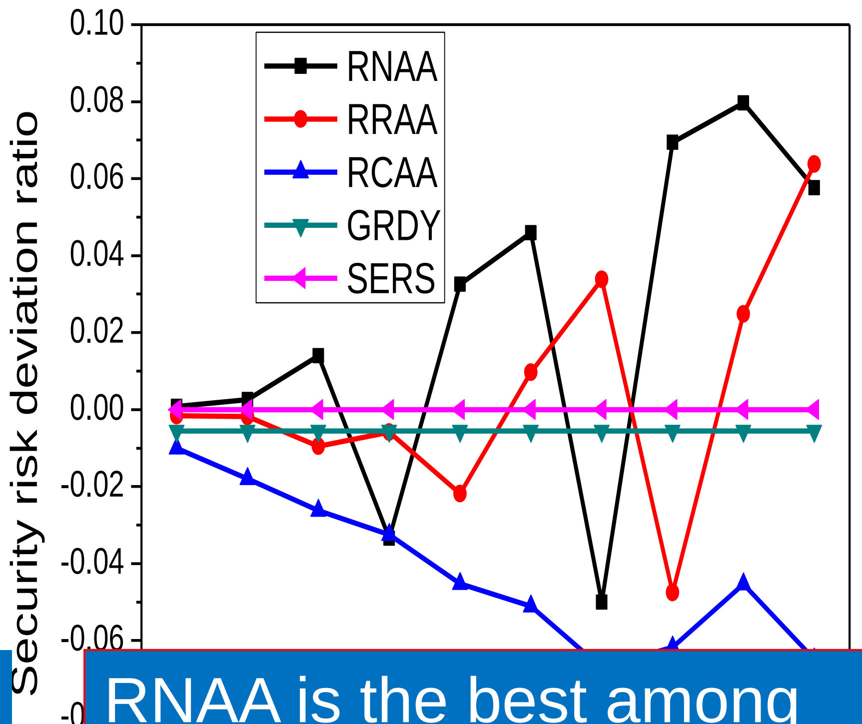

RNAA satisfies the risk slack ratio and its complexity is half of RCAA, RRAA.

# Impacts of Risk Slack Ratio ($\beta$)

- **Given** $RB = MIR + 0.7 * (MAR - MIR)$




RNAA saves 19.3%, 16.3%, 10% of GRDY, RCAA, SERS




RNAA is the best among them, lowest energy with guaranteed little risk deviation!

# Conclusions

- A new scheduling optimization problem for security- and energy-critical real-time applications
  - Minimal  energy with real-time and risk constraints
  - Multi-dimensional knapsack problem (NP-hard)

- Efficient techniques
  - Problem reduced (energy dimension)
  - Approximating dynamic programming
  - Half complexity of traditional approx. DP algorithms

- Experiments show the good performance

# Q&A

**Thanks for your time!**

**^_^**