A multi-task-oriented security processing architecture with powerful extensibility

1D-22

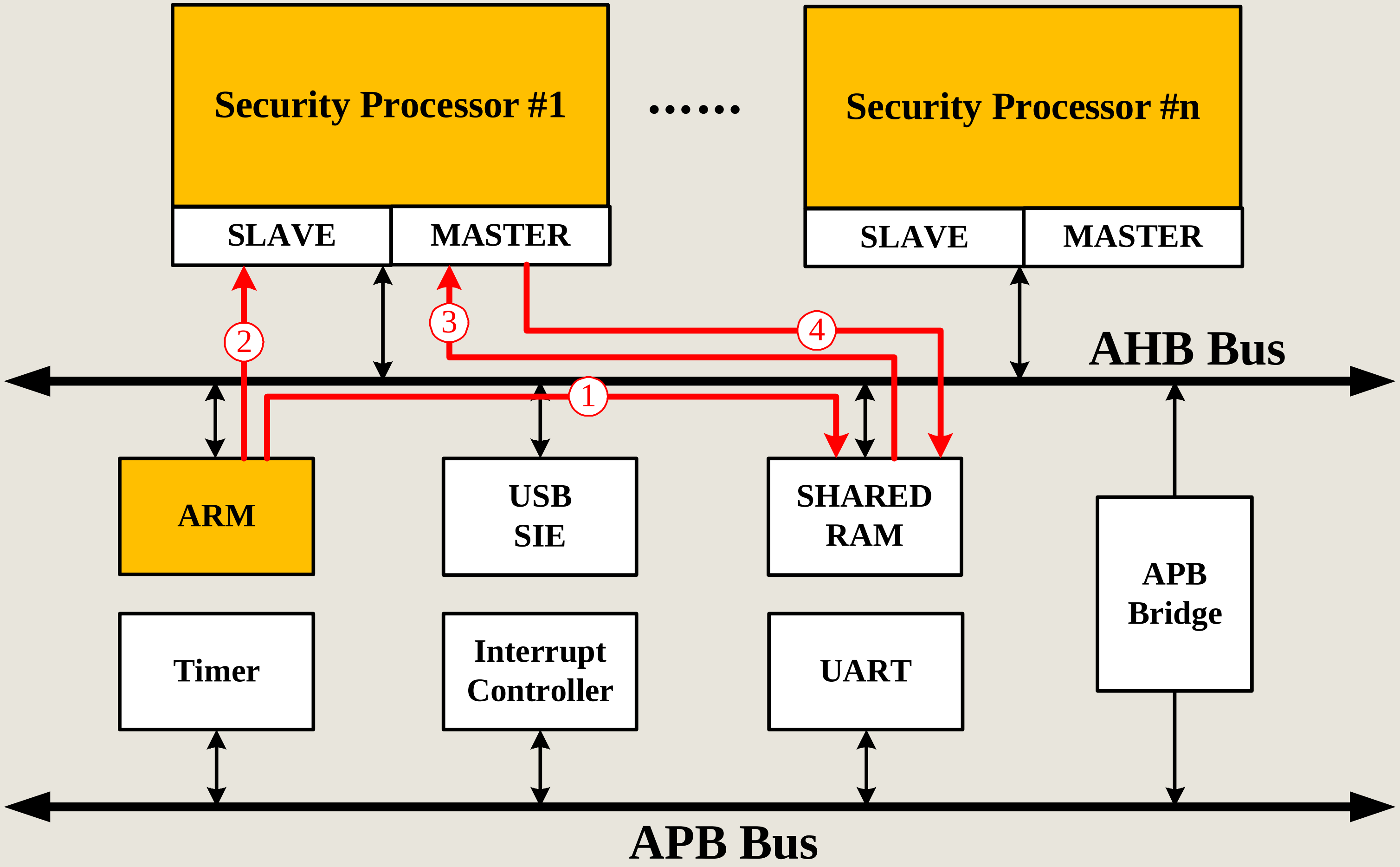
Dan Cao, Jun Han, Xiao-yang Zeng, Shi-ting Lu

Fudan University, China

Presented by Dan Cao

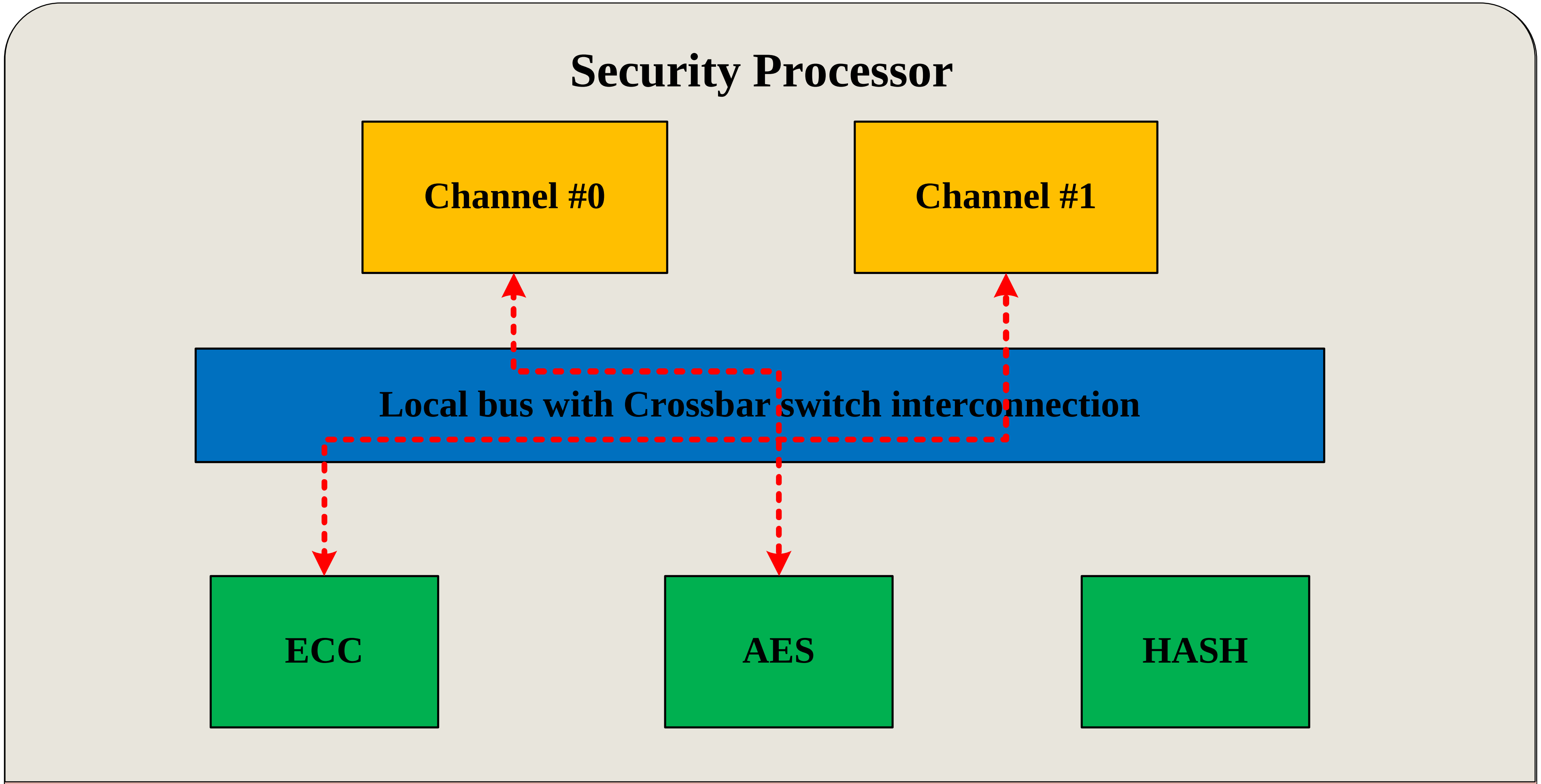
Motivation

- Function should be extensible for various cryptographic algorithms
- Processing ability should be extensible for flexible application demands
- More CPU resource should be reserved for other more complicated tasks



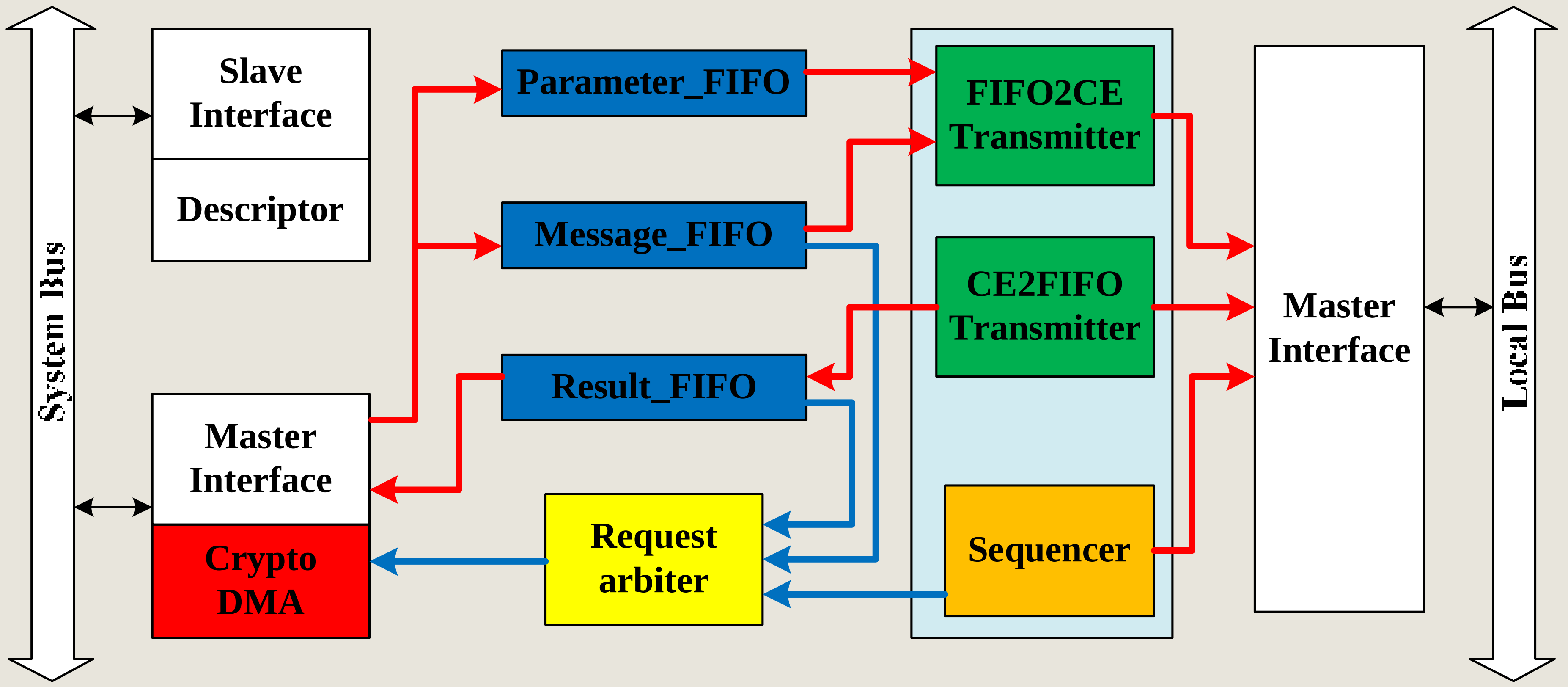
The architecture of the whole system

Host CPU + multiple SPs to achieve extensibility on processing ability



The design of Security Processor

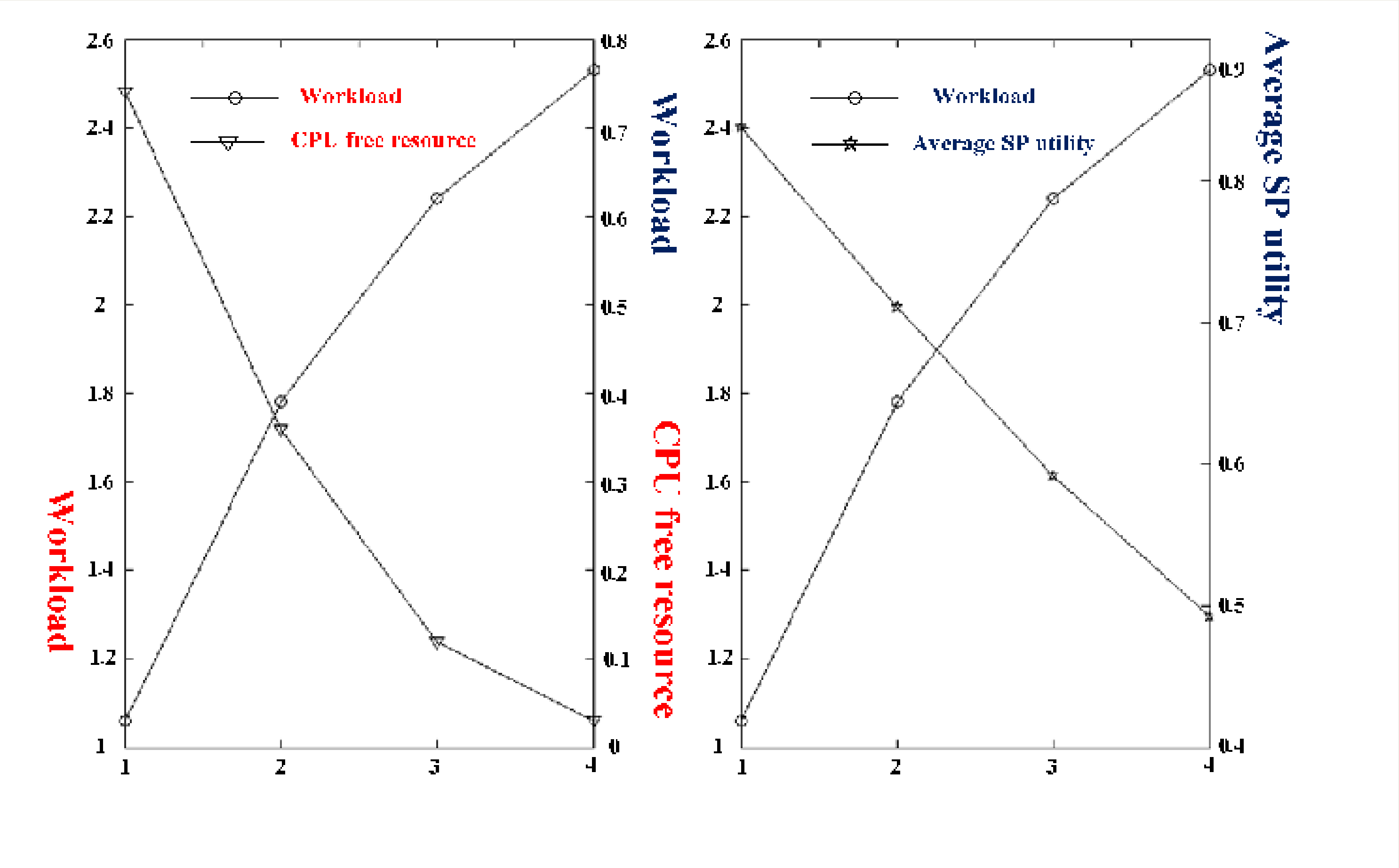
Task channels + local bus + Crypto-Engines to achieve functional scalability.



The diagram of task channel

Thread name	platform	Execution times per sec	CPU utility	SP utility
ECDSA (signature)	SLAVE	135	27%	N/A
	SP	140	5%	90%
ECDSA (authentication)	SLAVE	78	17%	N/A
	SP	80	3%	95%
ECDH (encryption)	SLAVE	186	5%	N/A
	SP	188	2%	97%
ECDH (decryption)	SLAVE	186	5%	N/A
	SP	188	2%	97%
CCMP	SLAVE	2267	99%	N/A
	SP	3422	42%	80%

Performance comparison between SP based SoC and SLAVE based SoC



The curves of the measurement indexes

A trade off can be made according to specific applications.

Conclusion

- Functional extensibility can be easily achieved through integrating new Crypto-Engines on Security Processor's local bus.
- Powerful processing ability can be obtained by placing more Security Processors on the system bus.
- However, a trade off between occupied CPU resource and performance should be made when multiple SPs are employed.