Programmable Crypto-Processors


Burak Erbagci


Ken Mai

While there are a wide variety of cryptographic algorithms, they tend to use only a relatively small set of well-understood functional primitives as building blocks. Among these operations are: data substitution through the substitution boxes (S-Box), modular addition/subtraction, Boolean functions (i.e. logical INV, OR, AND, XOR), and data re-arrangement such as rotation, shift or permutation. Although the specific type, amount, and ordering of these operations differ from one algorithm to algorithm, the algorithms have significant commonalities (Fig. 1). By exploiting these commonalities, our goal is to design a programmable crypto-processor architecture, which will support a broad range of cryptographic algorithms used for encryption, decryption, and authentication. The trade-off between flexibility (i.e., number of different algorithms supported) and performance (i.e., power/speed for each supported algorithm) determine the efficiency of the crypto processor compared to dedicated hardware solutions.

Among various industry-standard cryptographic algorithms, we selected a set of algorithms: AES, DES, SHA2, and some SHA3 candidates (Grøstl, Skein, JH and Blake) in order to determine the base architecture (Fig. 2) and exploitable parallelism. Then, we analyzed these algorithms and determined their architectural requirements such as the number and type of functional units as well as supporting instructions. We are currently working on the final processor architecture. Future work includes power/performance simulations of the final architecture, the design of a test chip prototype, and exploration of CAD tools for automated crypto-processor generation.

	AES	DES	SHA2	Blake	Skein	Grøstl	JH
SBOX	Yes 8x8	Yes 6x4	No	No	No	Yes 8x8	Yes 4x4
Add	No	No	Yes	Yes	Yes	No	No
Boolean Fn	XOR	XOR	XOR, AND	XOR	XOR	XOR	XOR
Data Arrange	Sh	Rot, Perm	Rot	Sh	Rot, Perm	Sh	Perm

Fig. 1: Commonalities between various cryptographic algorithms (Rot: rotation, Perm: permutation, Shft: shift)

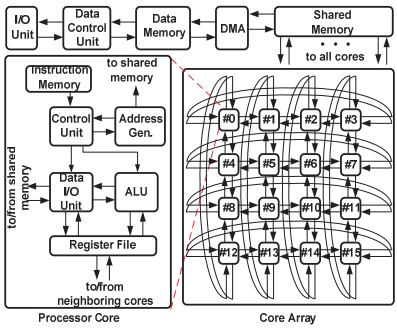

Fig. 2: The base architecture for selected set of algorithms (not all connections between blocks are shown, and 64b datapath is assumed)