Survey of Codebreaking Machines

Swathi Guruduth
Vivekanand Kamanuri
Harshad Patil

Contents

- Introduction
- Motivation
- Goal
- Machines considered
- Comparison based on technology used
- Brief description of machines
- FPGA machines
- ASICs and GPU
- Results
- Conclusions

Introduction

- Cryptography has been around for thousands of years
- Cryptanalysis is the study of breaking codes and ciphers
- Study of state of the art machines used for cryptanalysis

Motivation

- Useful for improvements for cryptosystem design
- Helpful to improve cryptosystem security

Goal

- Study of several parameters in cryptosystems and their comparison
- The project aims to give an overview of the possible hardware attacks on cryptosystems

Machines considered

- bcrypt password search (BPS) using special purpose hardware: 2014
- Cryptanalysis of full AES using GPU like hardware (CAESAR): 2012
- Cryptohaze GPU Rainbow (CGR) cracker: 2012
- Cryptanalysis of KeeLoq with COPACOBANA (CKC): 2010
- COPACOBANA: 2008
- NSA@home: 2007
- A fundamental evaluation of 80 bit keys employed by hardware oriented stream ciphers (EEK): 2006
- Implementing the elliptic curve method of factoring in reconfigurable hardware (ECMF): 2006

Comparison based on technology used

FPGA

- BPS using special purpose hardware: 2014
- Cryptanalysis of KeeLoq with COPACOBANA: 2010
- COPACOBANA: 2008
- NSA@home: 2007
- A fundamental evaluation of 80 bit keys employed by hardware oriented stream ciphers: 2006
- Implementing the ECM of factoring in reconfigurable hardware: 2006

ASICs

- CAESAR: 2012

GPUs

- Cryptohaze GPU Rainbow cracker: 2012

Brief description of machines

- BPS using special purpose hardware (2014)
  - This is a flexible high-speed implementation of a bcrypt password search system on a low-power Xilinx Zynq 7020 FPGA.
  - The design consists of 40 parallel bcrypt cores running at 100 Mhz.
  - This implementation outperforms all currently available implementations and improves password attacks on the same platform by at least 42%.

- CAESAR (2012)
  - Cryptanalysis of the full AES using GPU like hardware. It is a hypothetical supercomputer
  - The paper investigates the feasibility of large-scale hardware attacks on AES-128 and AES-256 bounded by a time complexity of 2^100, but memory complexity of less than 2^70 (and as little data as possible)

Contd..

- Cryptohaze GPU Rainbow cracker (2012)
  - The Cryptohaze tools are a set of GPU accelerated password cracking tools
  - Cryptohaze rainbow tables are a fully GPU accelerated implementation of the rainbow tables concept
  - The tools are cross platform and work with nVidia GPUs with CUDA, ATI GPUs with OpenCL, and both Intel & AMD CPUs with OpenCL

- Cryptanalysis of KeeLoq with COPACOBANA (2010)
  - In this paper a hardware architecture for the cryptanalysis of KeeLoq was developed
  - The brute-force attack, implemented on the cost-optimized parallel codebreaker COPACOBANA, is able to reveal the secret key of a remote control in less than 0.5 seconds if a 32-bit seed is used and in less than 6 hours in case of a 48-bit seed

Contd..

- COPACOBANA (2008)
  - Cost-optimized parallel code breaker is an FPGA-based machine which is optimized for running cryptanalytic algorithms
  - It is suitable for parallel computation problems which have low communication requirements

- NSA@home (2007)
  - NSA@home is a fast FPGA-based SHA-1 and MD5 brute-force cracker.
  - Capable of searching the full 8-character keyspace (from a 64-character set) in about a day in the current configuration for 800 hashes concurrently, using about 240W of power

Contd..

- A fundamental evaluation of 80 bit keys employed by hardware oriented stream ciphers (2006)
  - In this paper the security afforded by the 80 bit keys of hardware focused stream cipher is analyzed from the perspective of brute force attack susceptibility

- Implementing the ECM of factoring in reconfigurable hardware (2006)
  - A novel hardware architecture for the ECM of factoring has been proposed
  - The ECM architecture has been ported across five different families of FPGA devices in order to select the family with the best performance to cost ratio
  - A timing comparison with the highly optimized software implementation, GMP-ECM, has been performed

FPGA Machines

Machine	Technology	type of codes it can crack	Time	costs	power
CKC	Cluster COPACOBANA (Xilinx Spartan 3 FPGAs)	Block Ciphers	Less than 0.5sec for a 32-bit seed and less than 6hrs in case of a 48-bit seed	Less than $10,000	600W
ECMF	Xilinx Spartan 3 FPGA	RSA Factorization	33.5msec using a Spartan 3 XC3S5000-5	$130	
COPACOBANA	Xilinx Spartan 3 with microblaze	Any symmetric cipher with up to roughly 64 key bits	6.4 days	Less than $10,000	600W

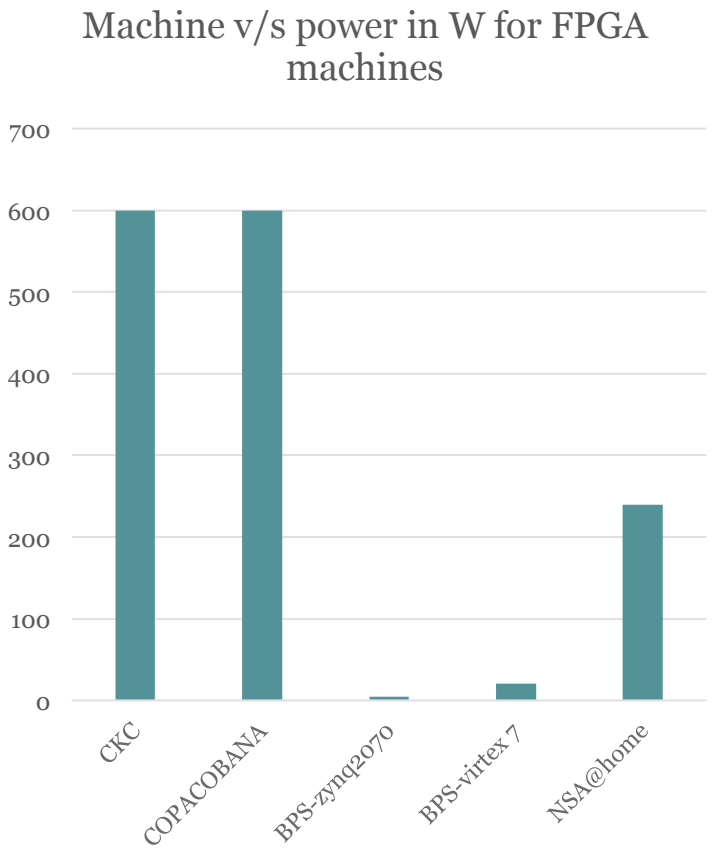
Contd..

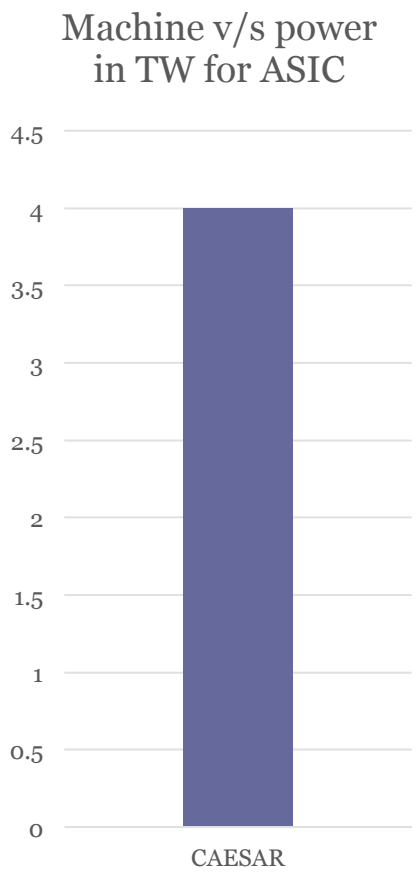
Machine	Technology	type of codes it can crack	Time	costs	power
bcrypt	Zedboard, Xilinx Zynq 7020 FPGA	bcrypt passwords	6,511 hashes per sec	$319	4.2W
	Virtex 7		51,437 hashes per sec	$3,495	20W
NSA@home	FPGA Virtex II pro	hash function	8 character keyspace in about a day		240W
EEK	Altera's Cyclone II and HardCopy II	Stream Ciphers	80 bit keys in 1hr using EP2C35 FPGA and in 1min using HC210 FPGA	Cyclone II - $68 billion and for HardCopy II - $240 billion by 2015	Cyclone II approx. 180 mW and 360 mW for on-chip system and off-chip respectively

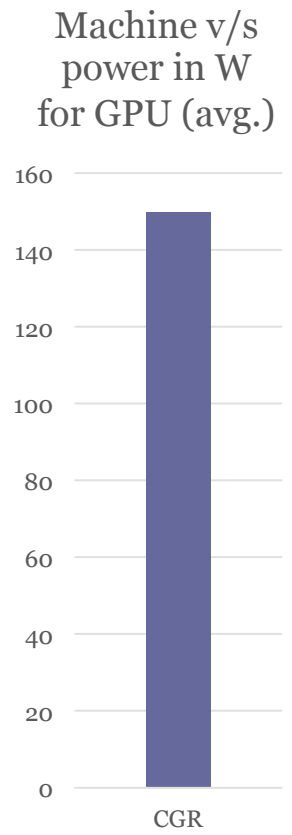
ASICs and GPU

Machine	Technology	Type of codes it can crack	Time	Costs	Power
CAESAR	ASIC	AES operations	9*10^29 AES operations can be done in 3*10^7secs	Expected to be $1 trillion	4 TW
CGR	GPUs and Open CL	Passwords	A hash function can be cracked in under 2mins	Cost of a personal computer (Less than $1000)	60W to 250W

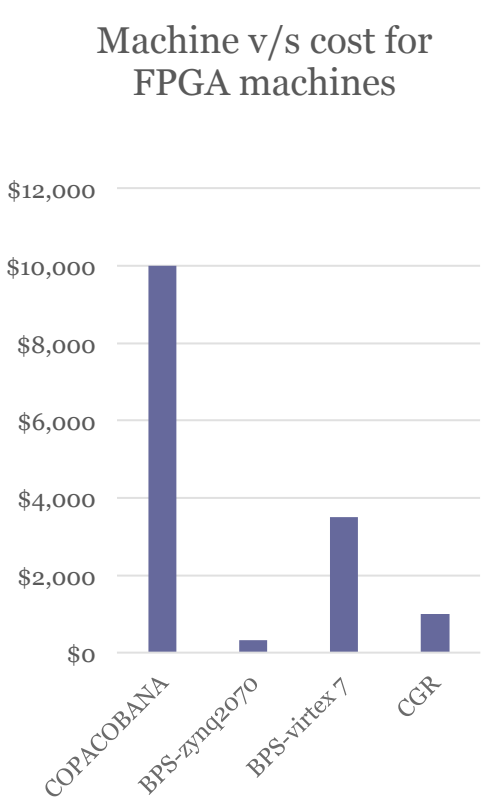
Results: Power

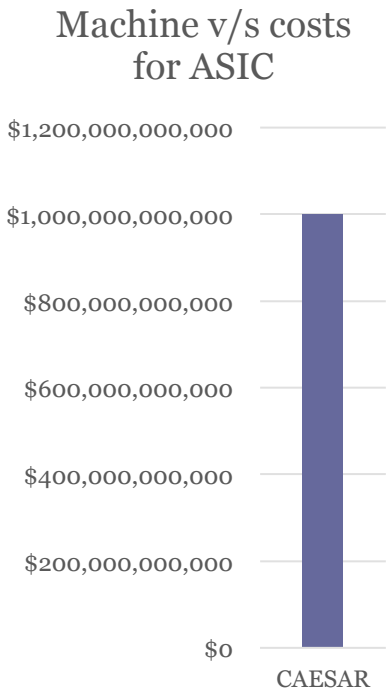


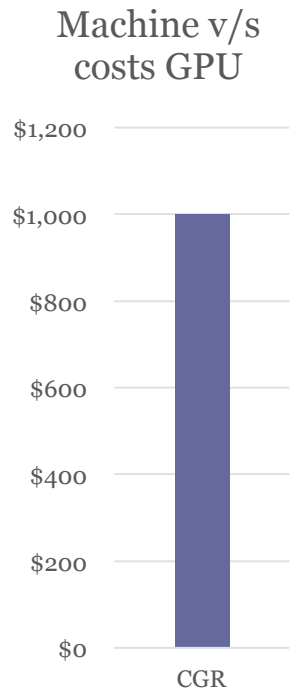




Results: Costs

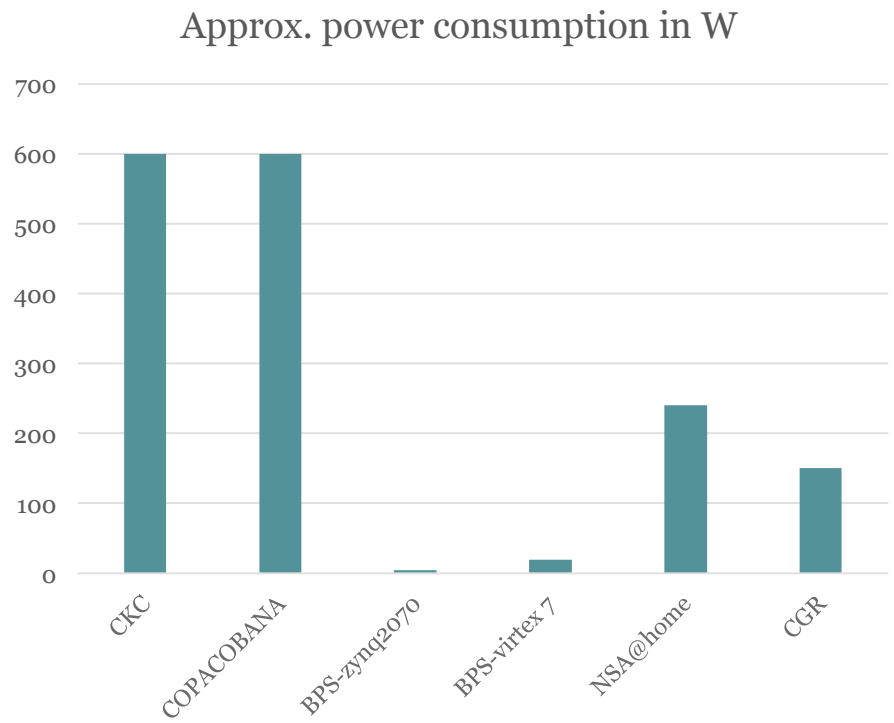






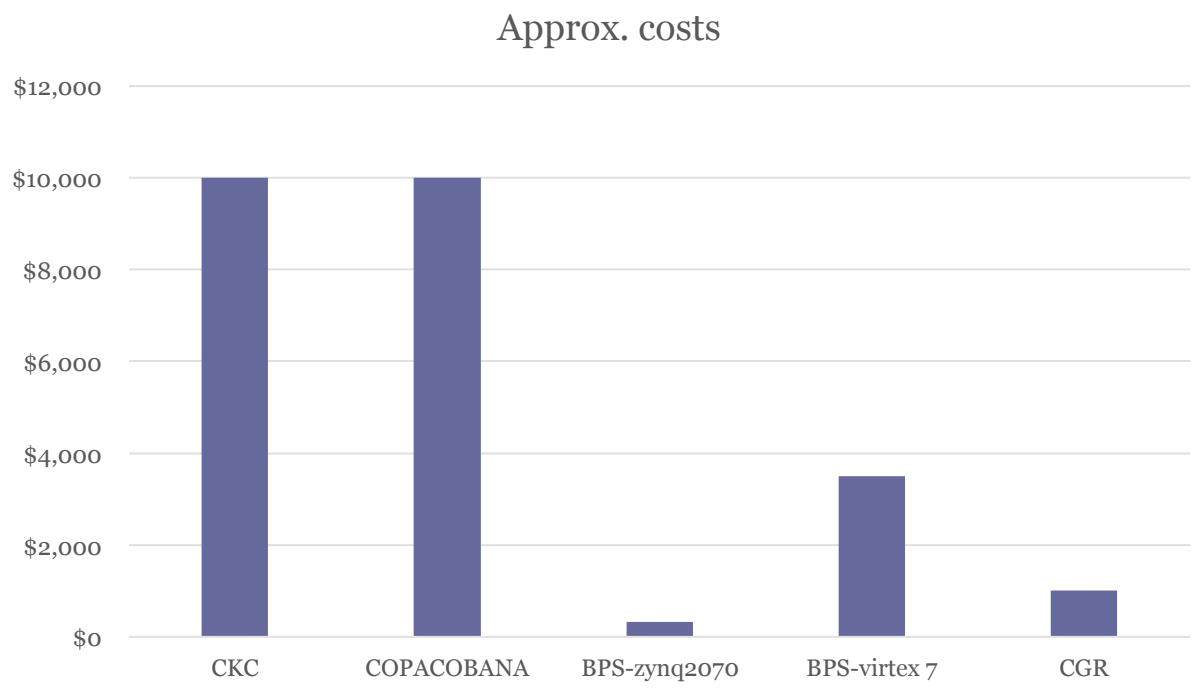
Conclusions

- Altera's cyclone II and HardCopy II are the machine which uses minimum power to break stream ciphers.
- CAESAR consumes most power: 4TW



Conclusions

- Implementing the Elliptic Curve Method of Factoring on a Spartan 3 FPGA is the most cost effective at $130
- Least cost effective is CEASAR costing close to 1 trillion USD




THANK YOU