OpenCiphers

Cracking WiFi… Faster!

*(**F**aster **P**wnin**G** **A**ssured)*

RECON 2006 – June 16th, 2006

David Hulton <dhulton@openciphers.org>

Cracking WiFi… Faster!

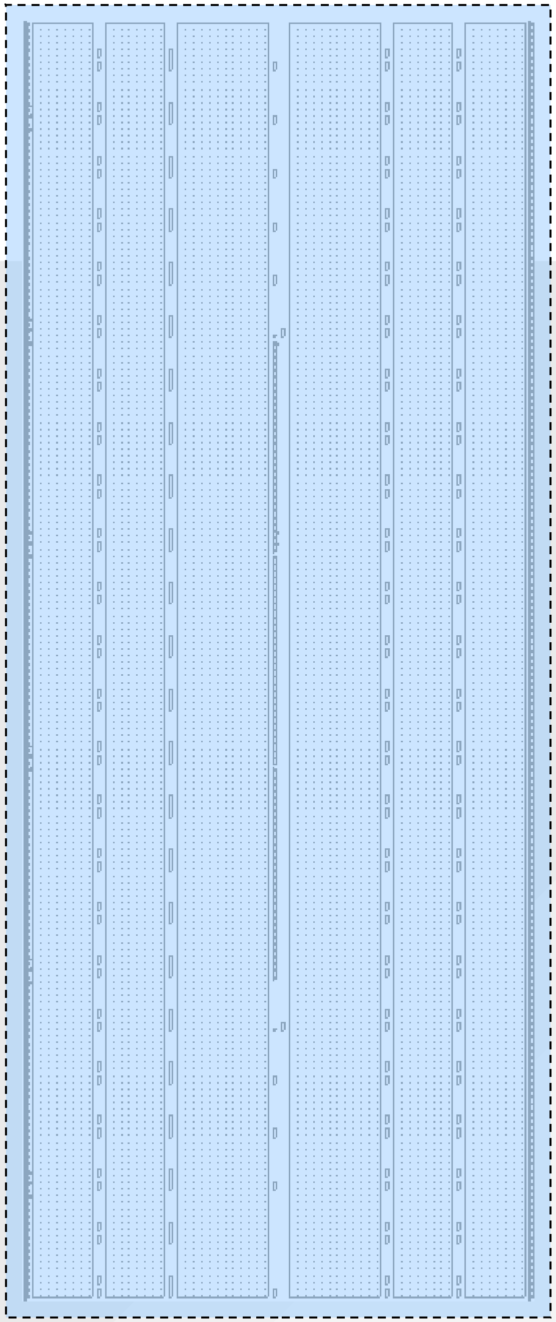
- FPGAs
  - Quick Intro (I swear!)
- coWPAtty
  - WPA Overview
  - Precomputing tables
  - Performance
- Airbase
  - jc-aircrack
  - jc-wepcrack
  - pico-wepcrack
  - Performance
- Conclusion

FPGAs

- Quick Intro
  - Chip with a ton of general purpose logic
    - ANDs, ORs, XORs
    - FlipFlops (Registers)
    - BlockRAM (Cache)
    - DSP48's (ALUs)
    - DCMs (Clock Multipliers)

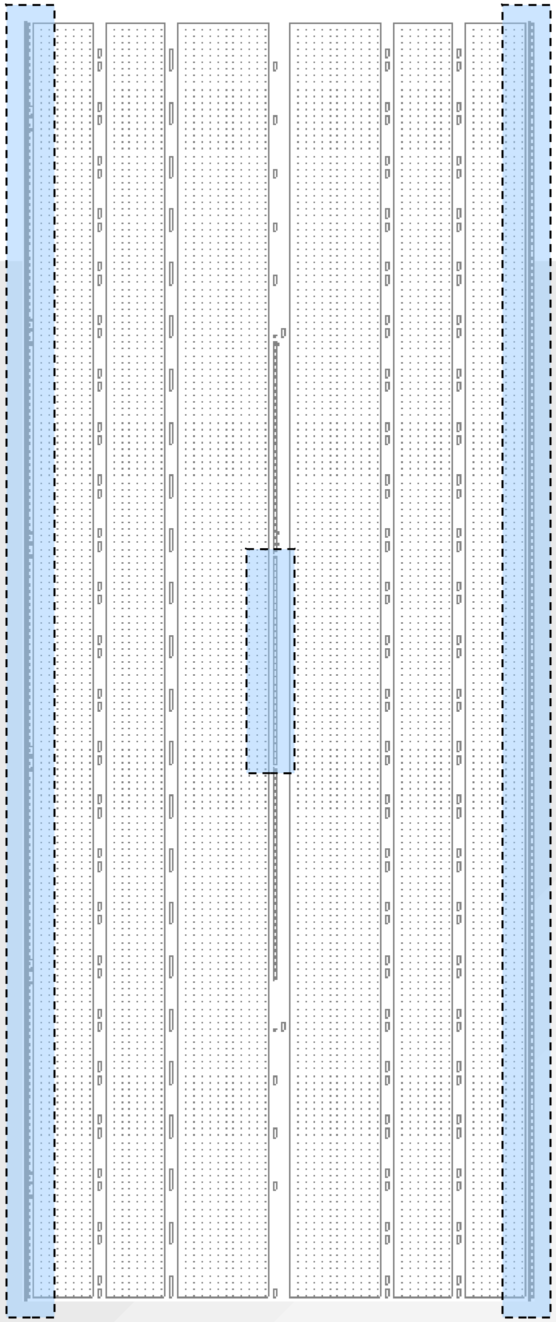
FPGAs

- Virtex-4 LX25



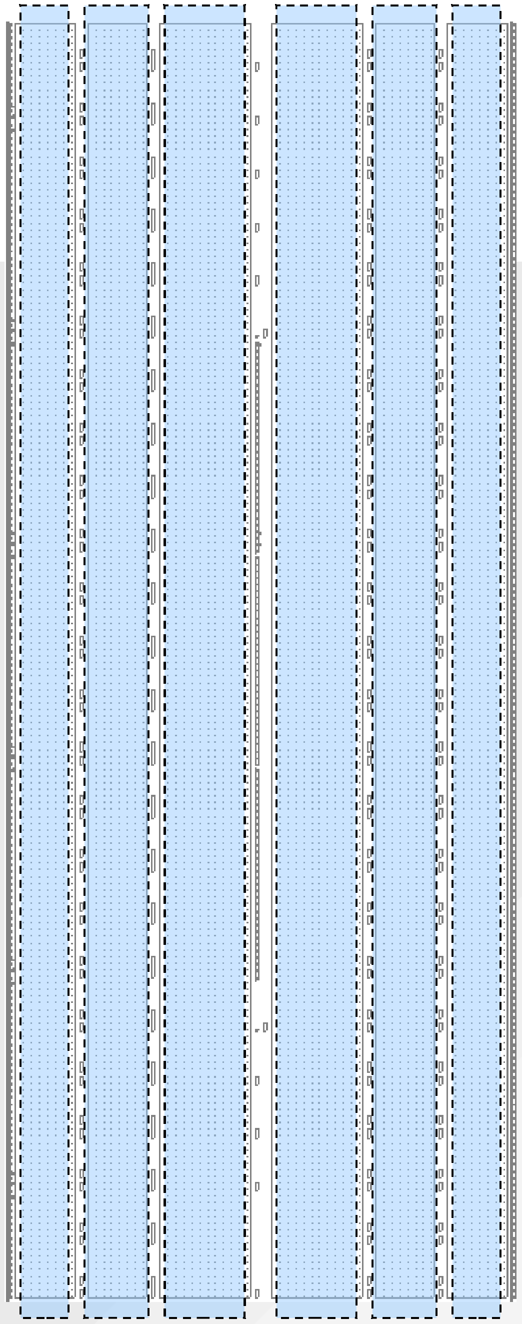
FPGAs

- Virtex-4 LX25
  - IOBs (448)



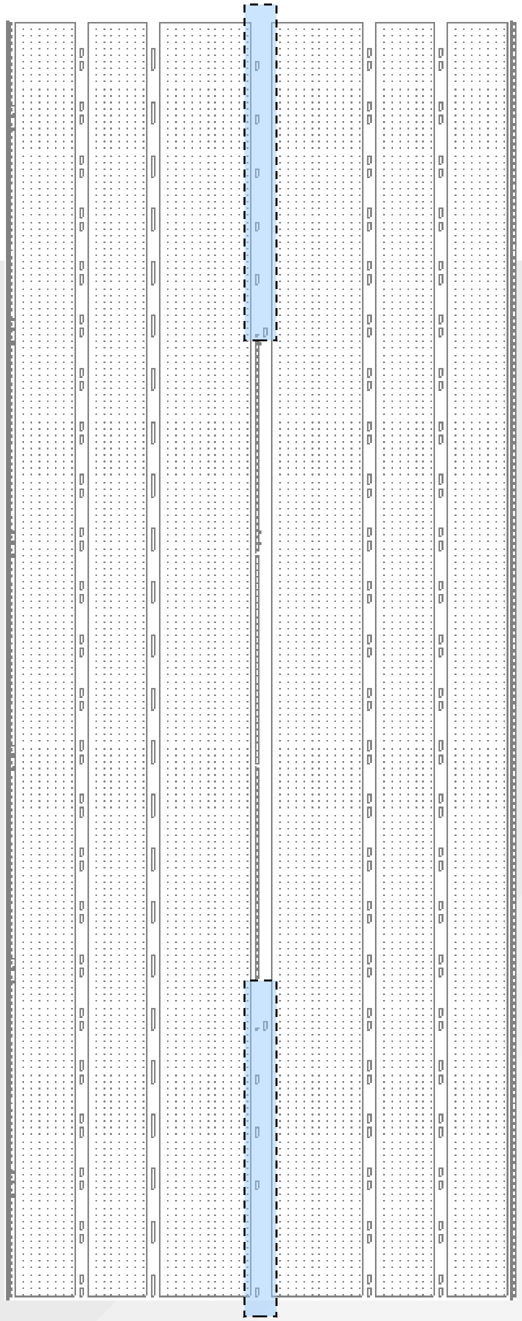
FPGAs

- Virtex-4 LX25
  - IOBs
  - Slices (10,752)



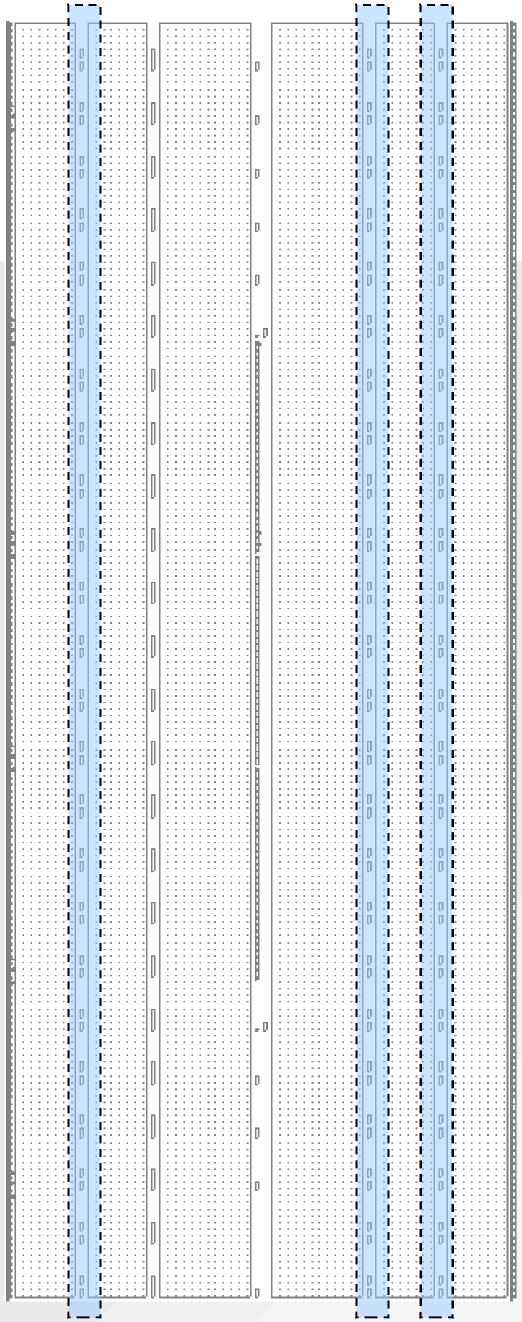
FPGAs

- Virtex-4 LX25
  - IOBs
  - Slices
  - DCMs (8)



FPGAs

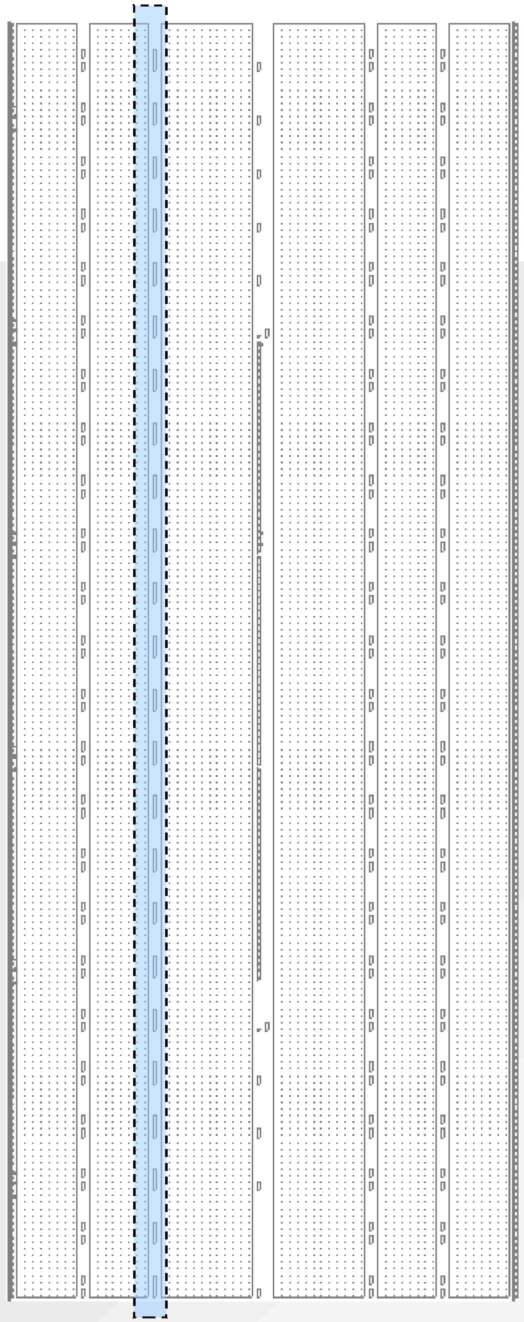
- Virtex-4 LX25
  - IOBs
  - Slices
  - DCMs
  - BlockRAMs (72)



OpenCiphers

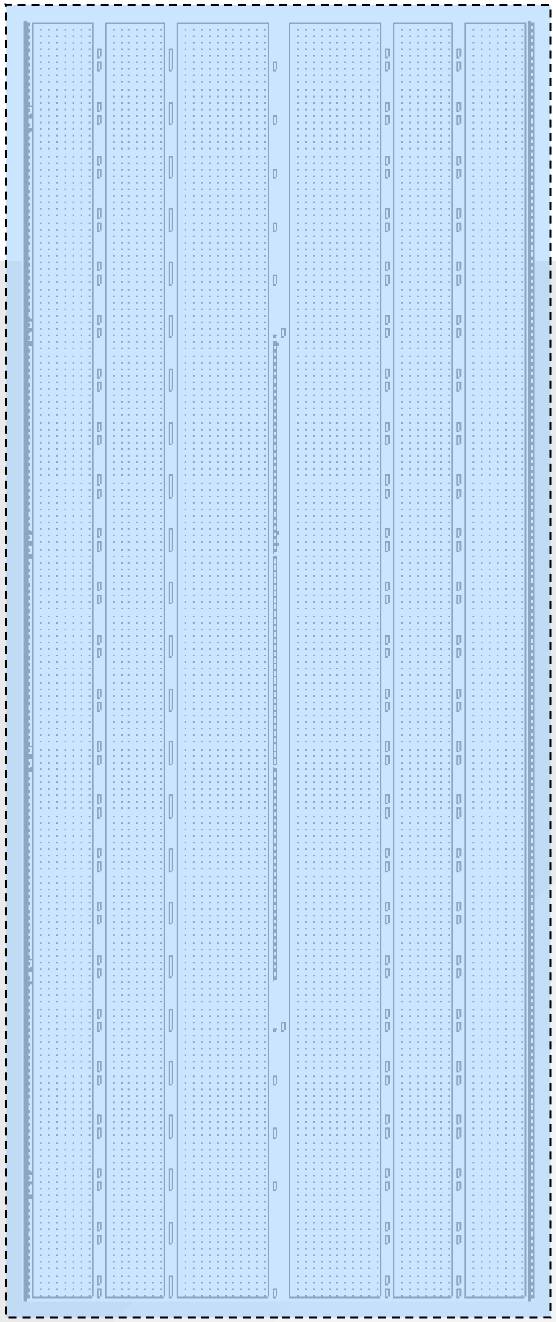
FPGAs

- Virtex-4 LX25
  - IOBs
  - Slices
  - DCMs
  - BlockRAMs
  - DSP48s (48)



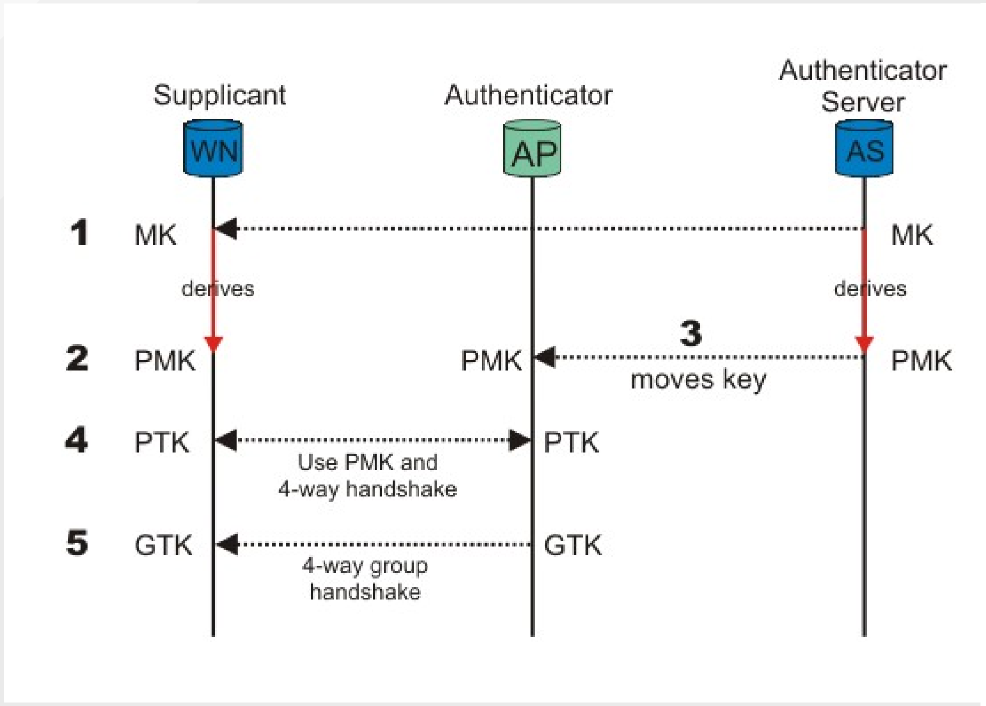
FPGAs

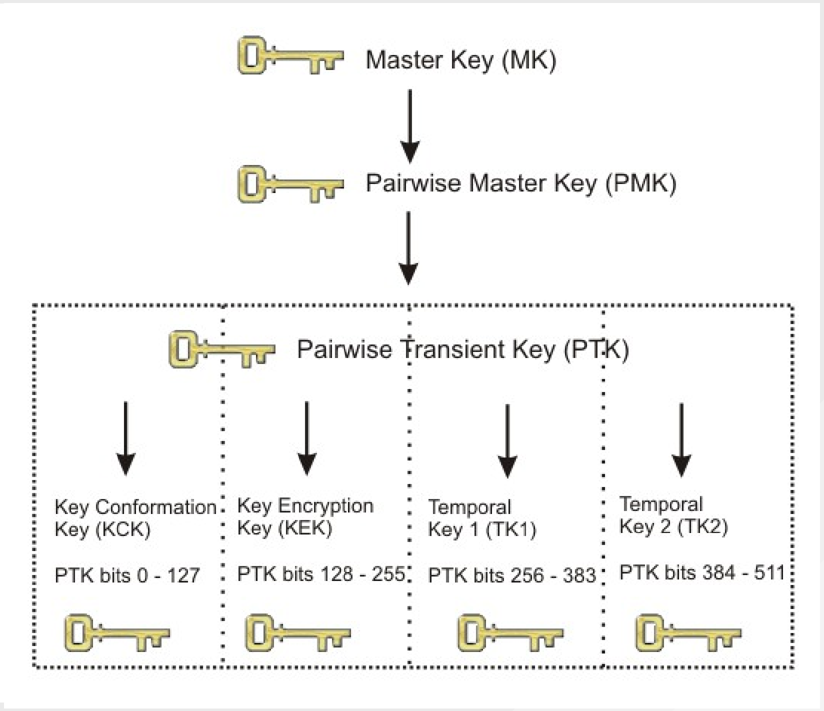
- Virtex-4 LX25
  - IOBs
  - Slices
  - DCMs
  - BlockRAMs
  - DSP48s
  - Programmable Routing Matrix (~18 layers)



Introduction to WPA

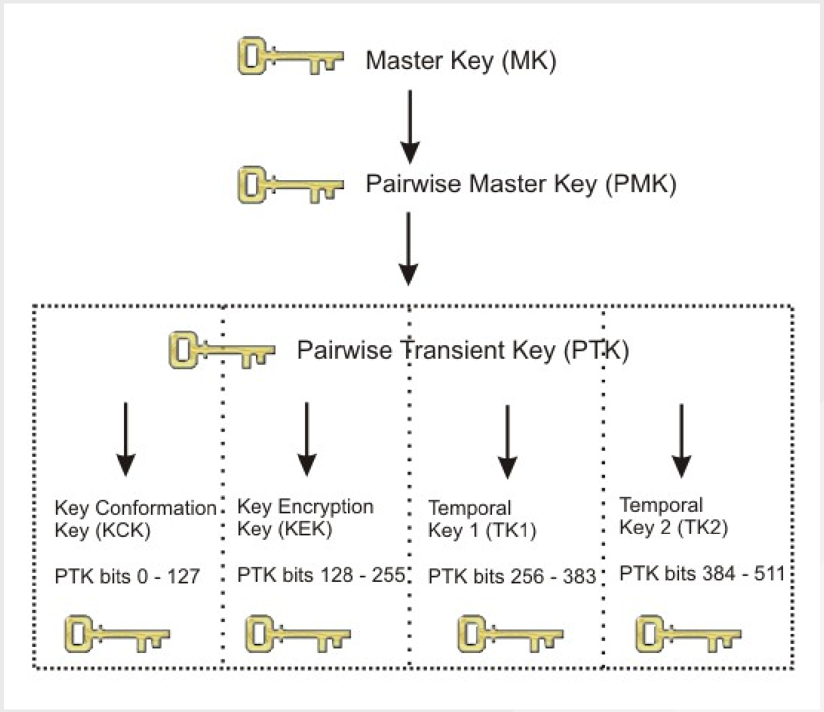
- WiFi Protected Access





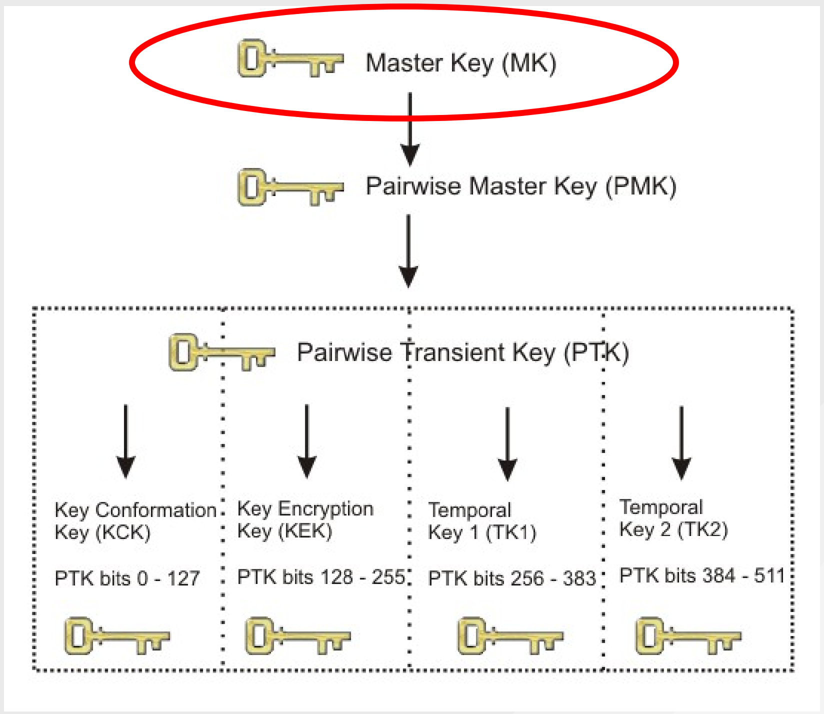
Introduction to WPA

- PSK
  - MK is your passphrase
  - It’s run through PBKDF2 to generate the PMK



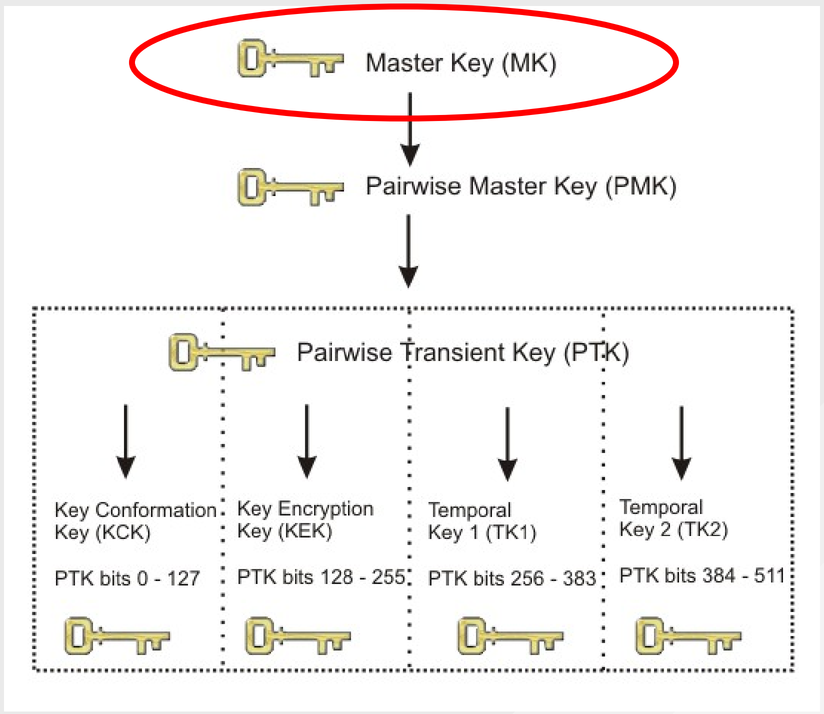
Introduction to WPA

- PSK
  - MK is your passphrase
  - It’s run through PBKDF2 to generate the PMK



Introduction to WPA

- PSK
  - MK is your passphrase
  - It's run through PBKDF2 to generate the PMK



Introduction to WPA

- PBKDF2

```
unsigned char hash[32];

t = sha1_hmac(MK, SSID, 1);
for(i = 1; i < 4096; i++)
    t = sha1_hmac(MK, t);
memcpy(hash, &t, 20);

t = sha1_hmac(MK, SSID, 1);
for(i = 1; i < 4096; i++)
    t = sha1_hmac(MK, t);
memcpy(hash + 20, &t, 12);
```

Introduction to WPA

- sha1_hmac

```
sha1(MK ^ 0x5c, sha1(MK ^ 0x36, t));
```

```
sha1init(ctx);
ctx = sha1update(ctx, MK ^ 0x36);
ctx = sha1update(ctx, t);
innersha1_ctx = sha1final(ctx);

sha1init(ctx);
ctx = sha1update(ctx, MK ^ 0x5c);
ctx = sha1update(ctx, innersha1_ctx);
outersha1_ctx = sha1final(ctx);
```

OpenCiphers

Introduction to WPA

- sha1_hmac

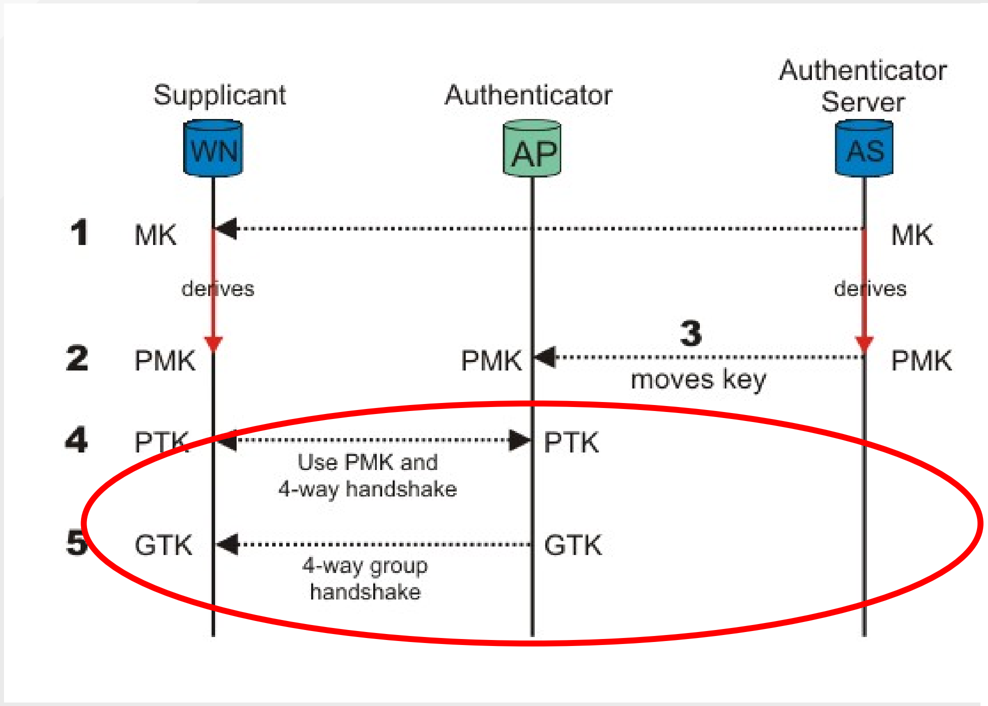
```
sha1(MK ^ 0x5c, sha1(MK ^ 0x36, t));
```

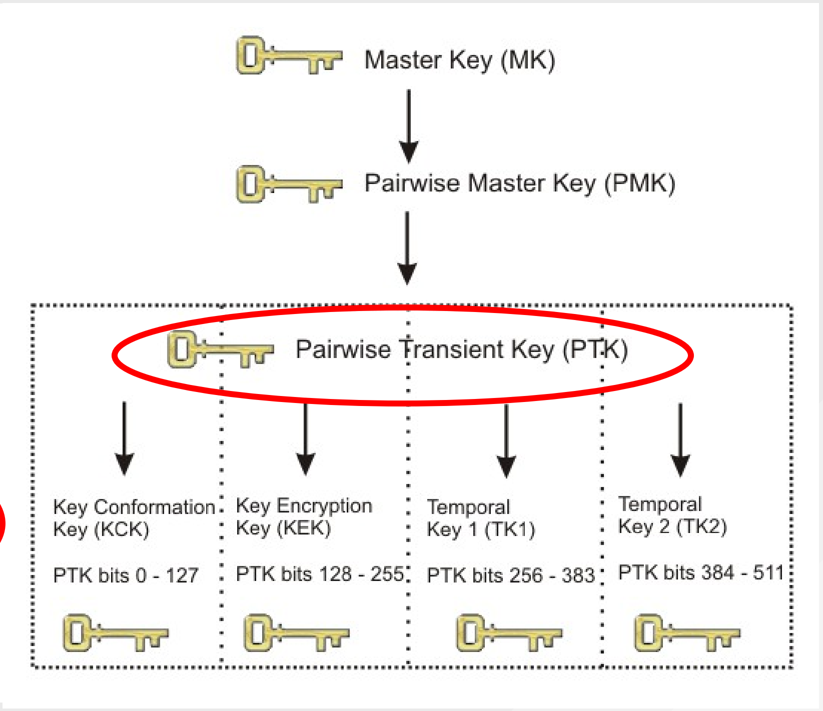
```
sha1init(ctx);
ctx = sha1update(ctx, MK ^ 0x36);
ctx = sha1update(ctx, t);
innersha1_ctx = sha1final(ctx);

sha1init(ctx);
ctx = sha1update(ctx, MK ^ 0x5c);
ctx = sha1update(ctx, innersha1_ctx);
outersha1_ctx = sha1final(ctx);
```

You can cache
some of the state
to reduce the number
of required SHA1's

Introduction to WPA

- For every possible PMK compute PTK and see if it matches the handshake captured on the network

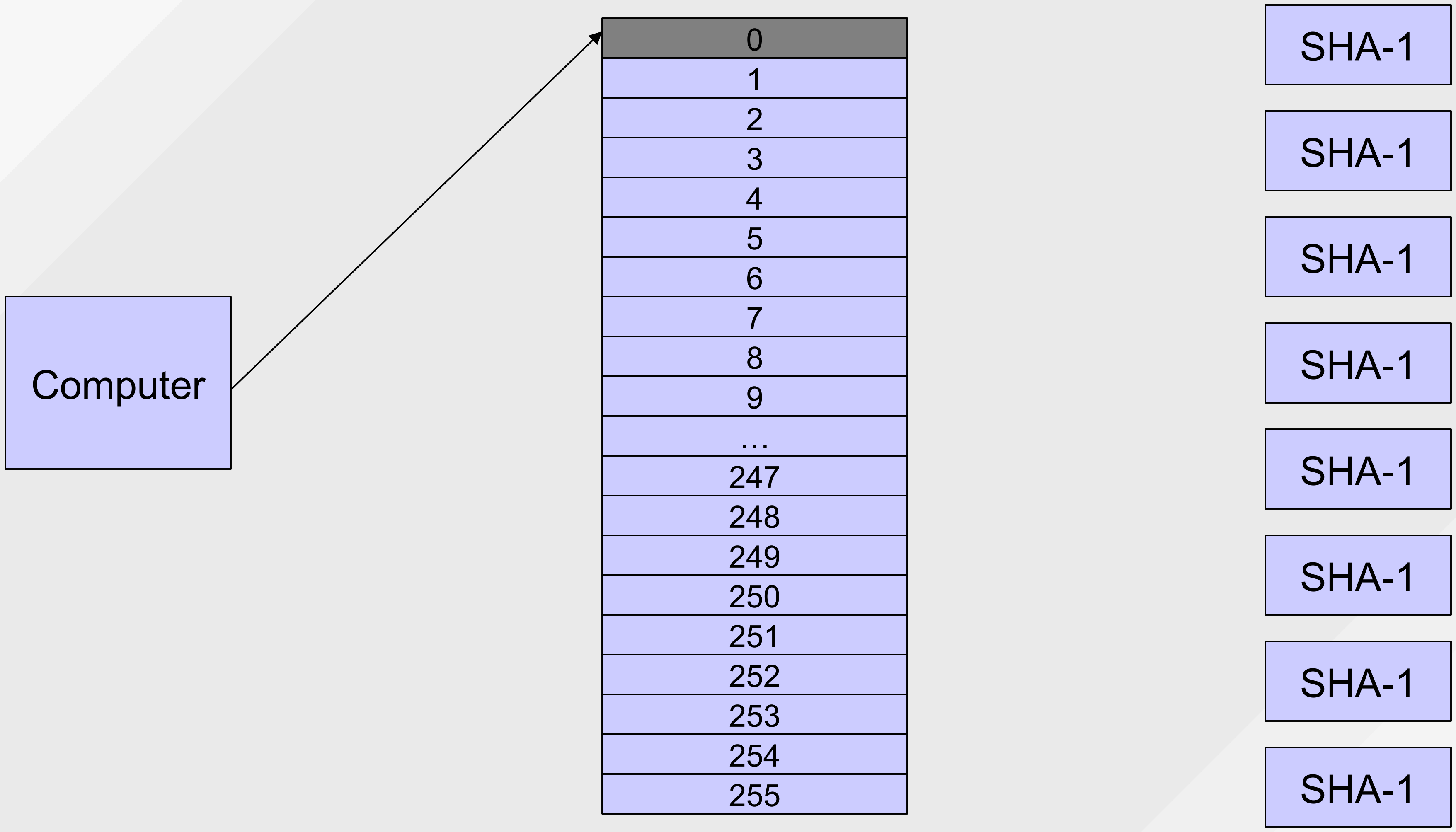




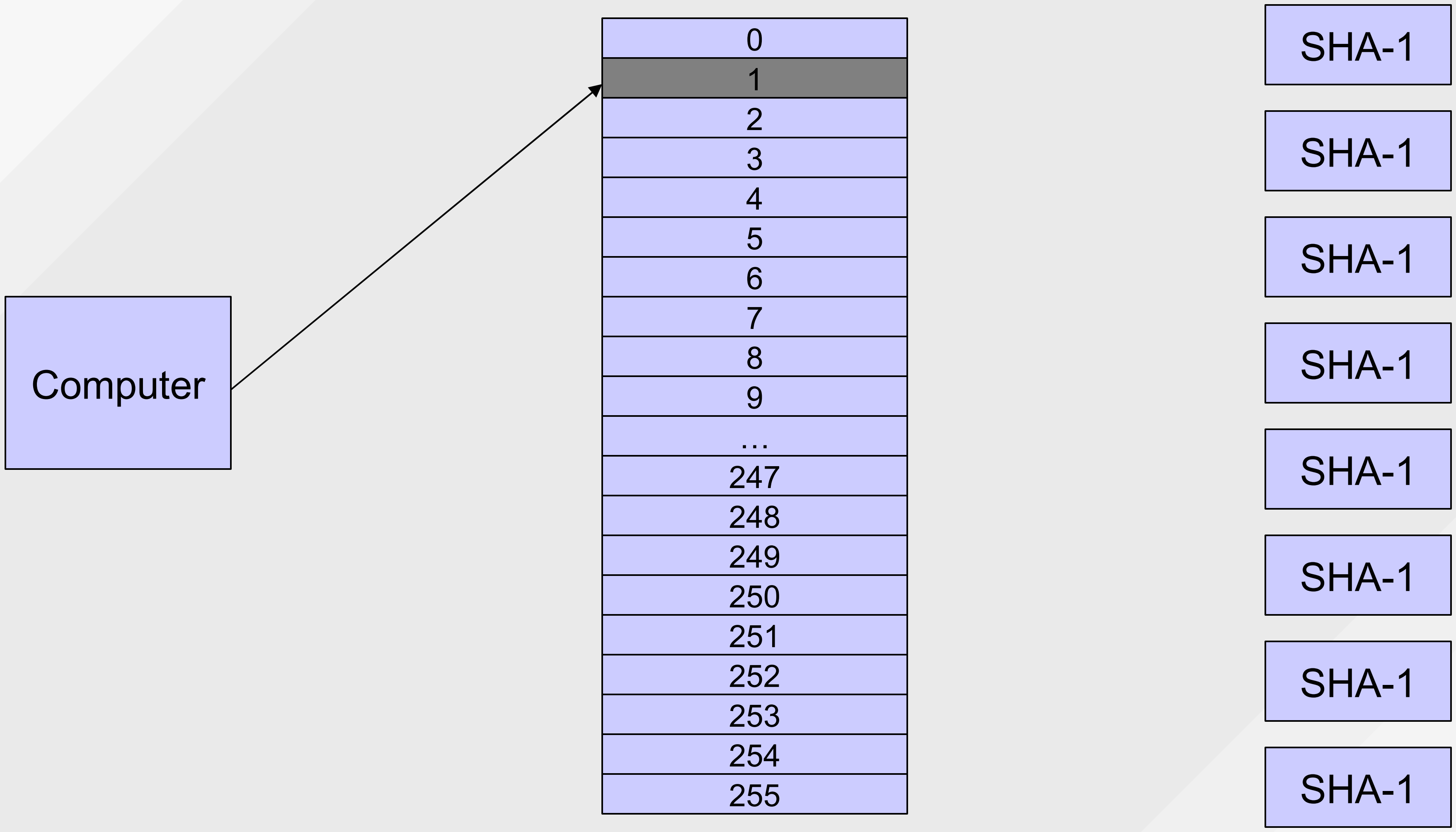
FPGA coWPAtty

- Uses 8 SHA-1 Cores
- Uses BlockRAM to buffer the words fed to the cores
- As long as the machine is able to supply words fast enough, the SHA-1 cores will be utilized fully

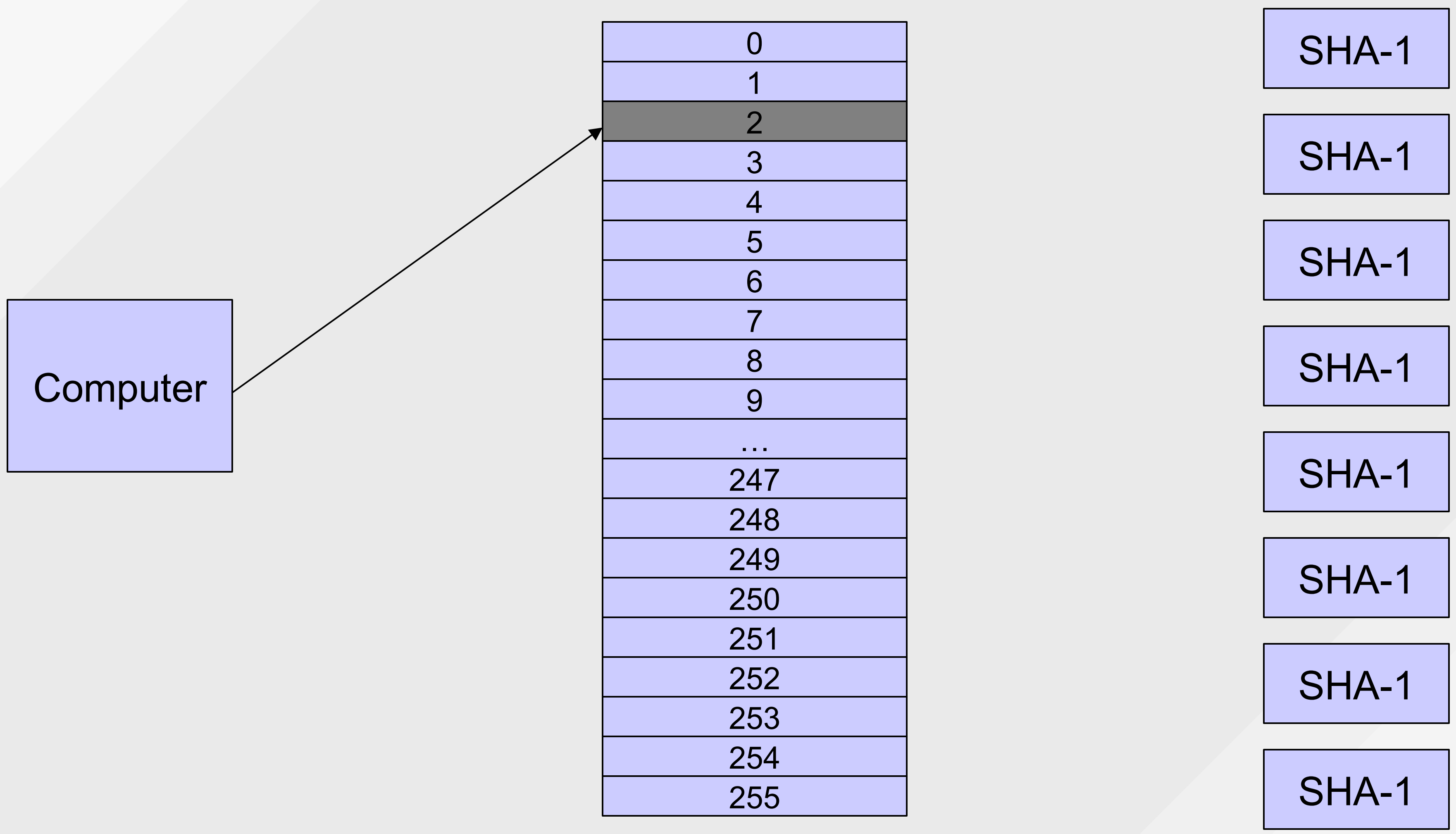
FPGA coWPAtty



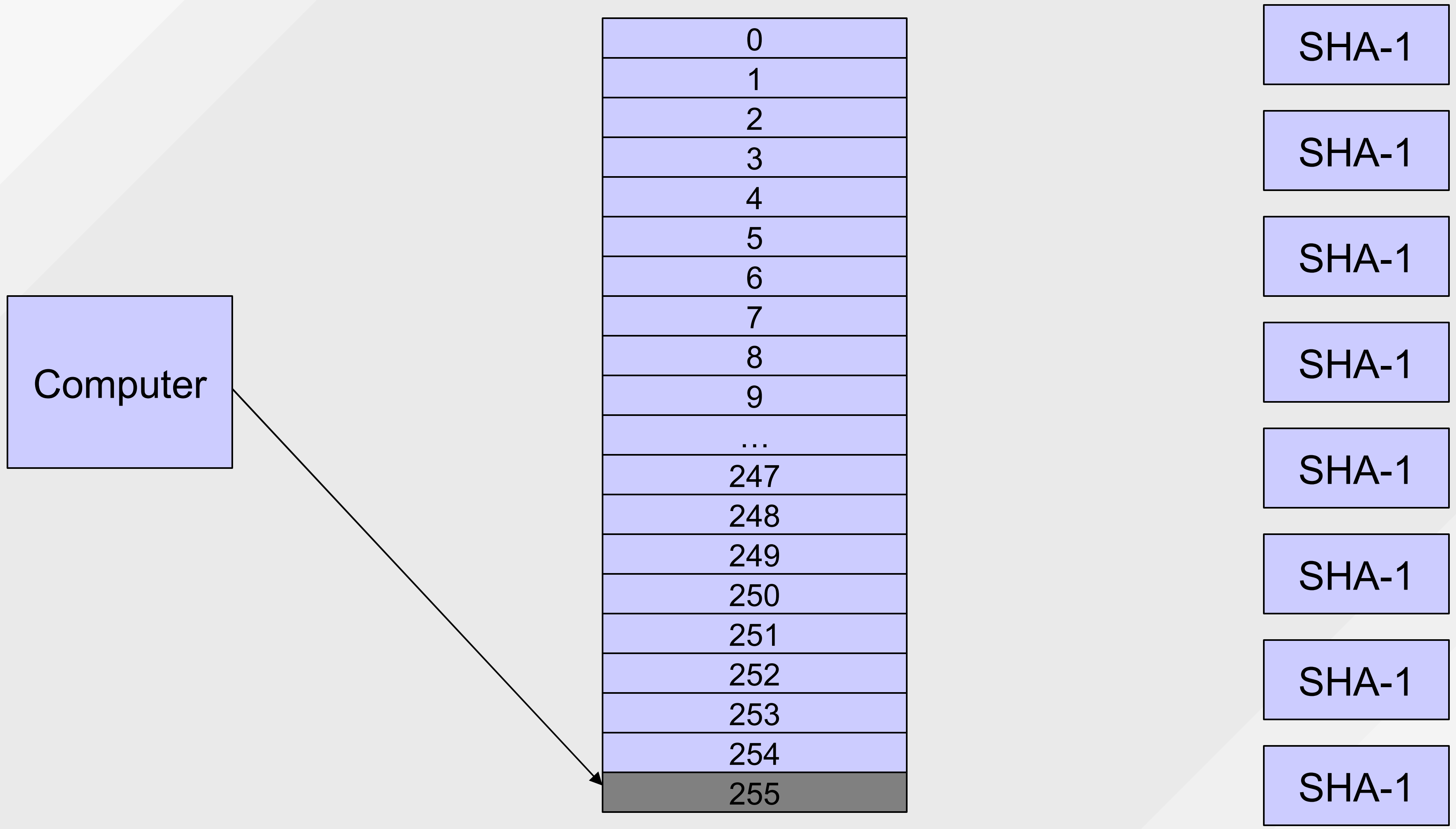
FPGA coWPAtty



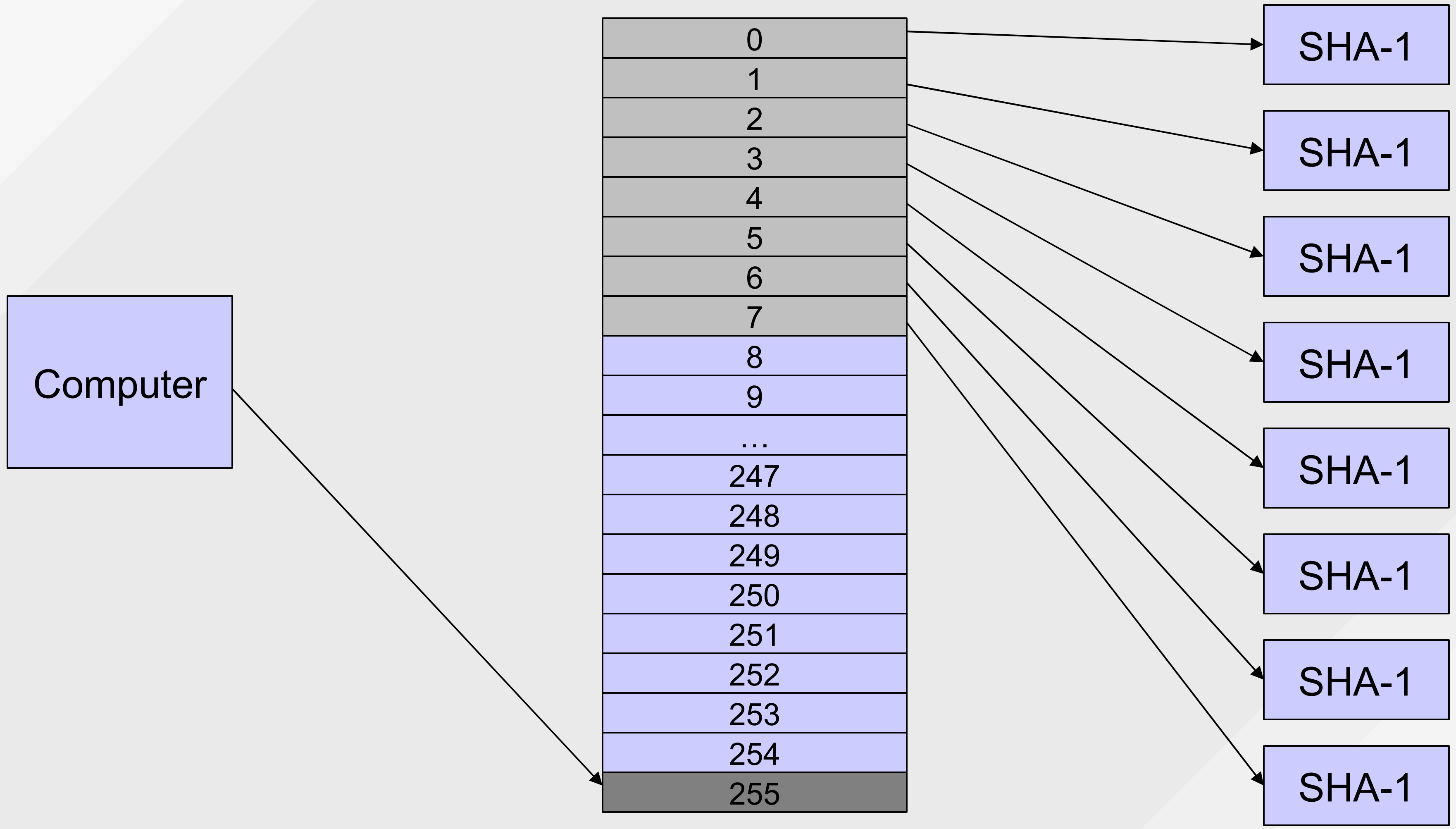
FPGA coWPAtty



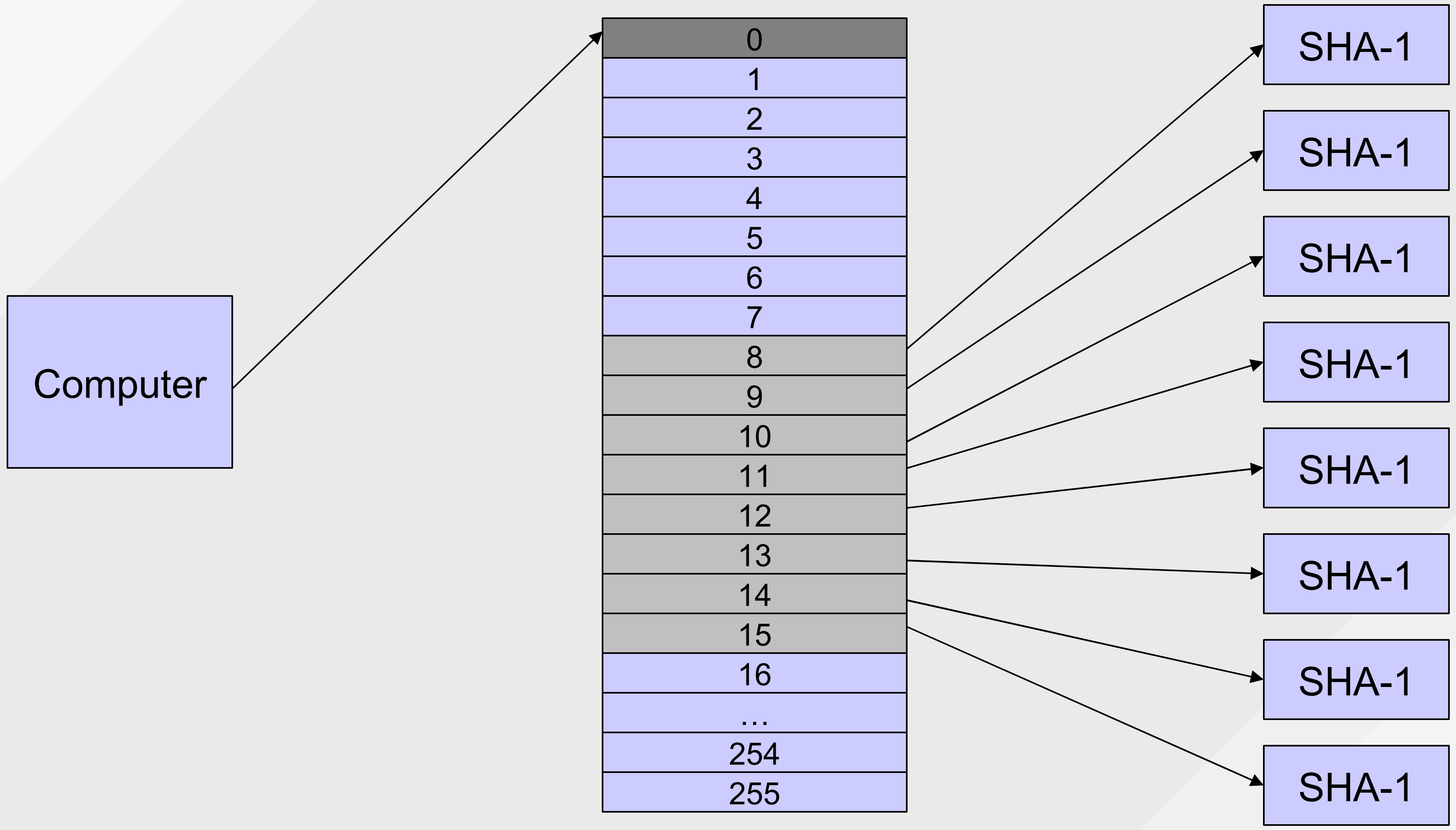
FPGA coWPAtty



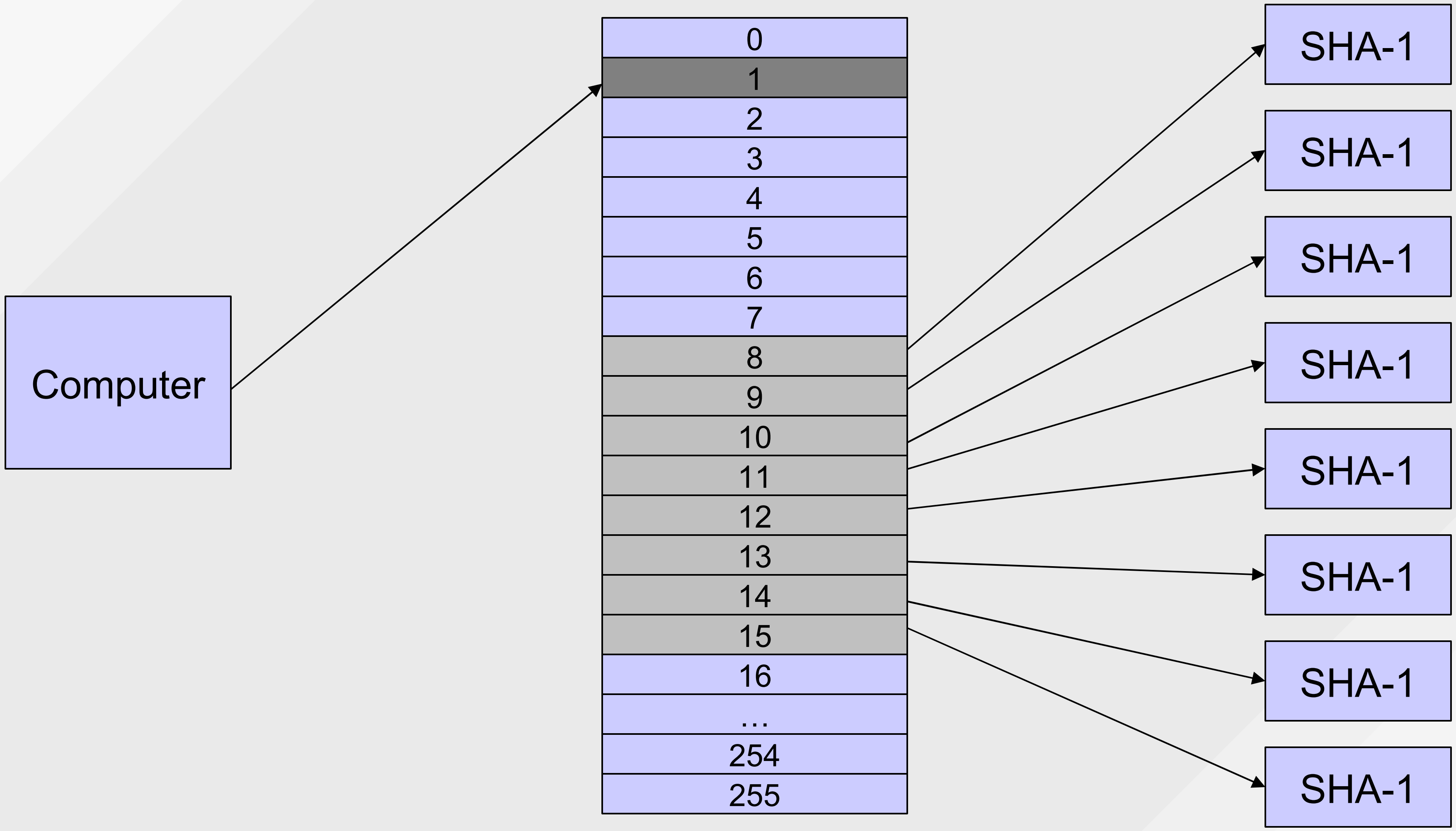
FPGA coWPAtty



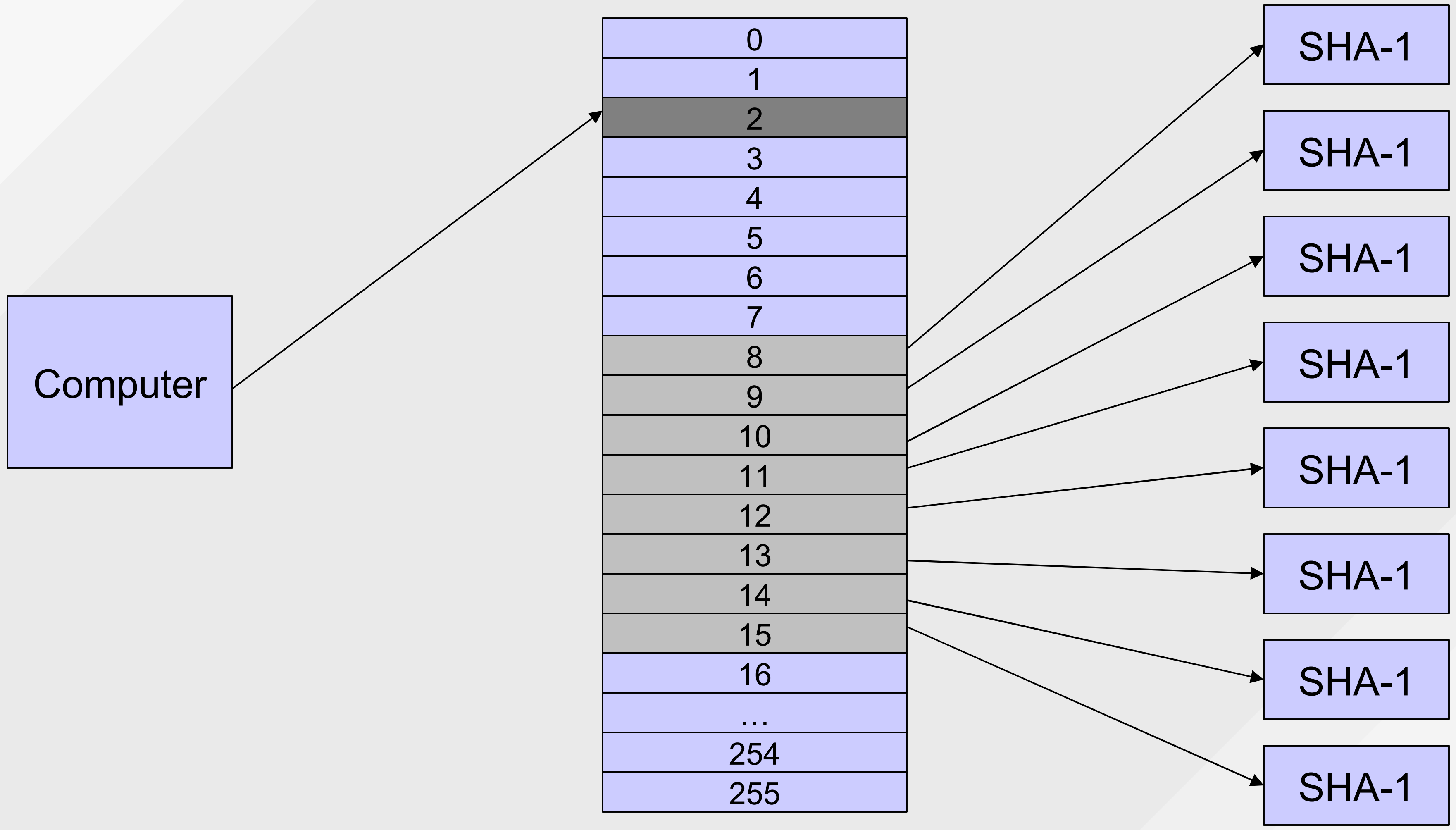
FPGA coWPAtty



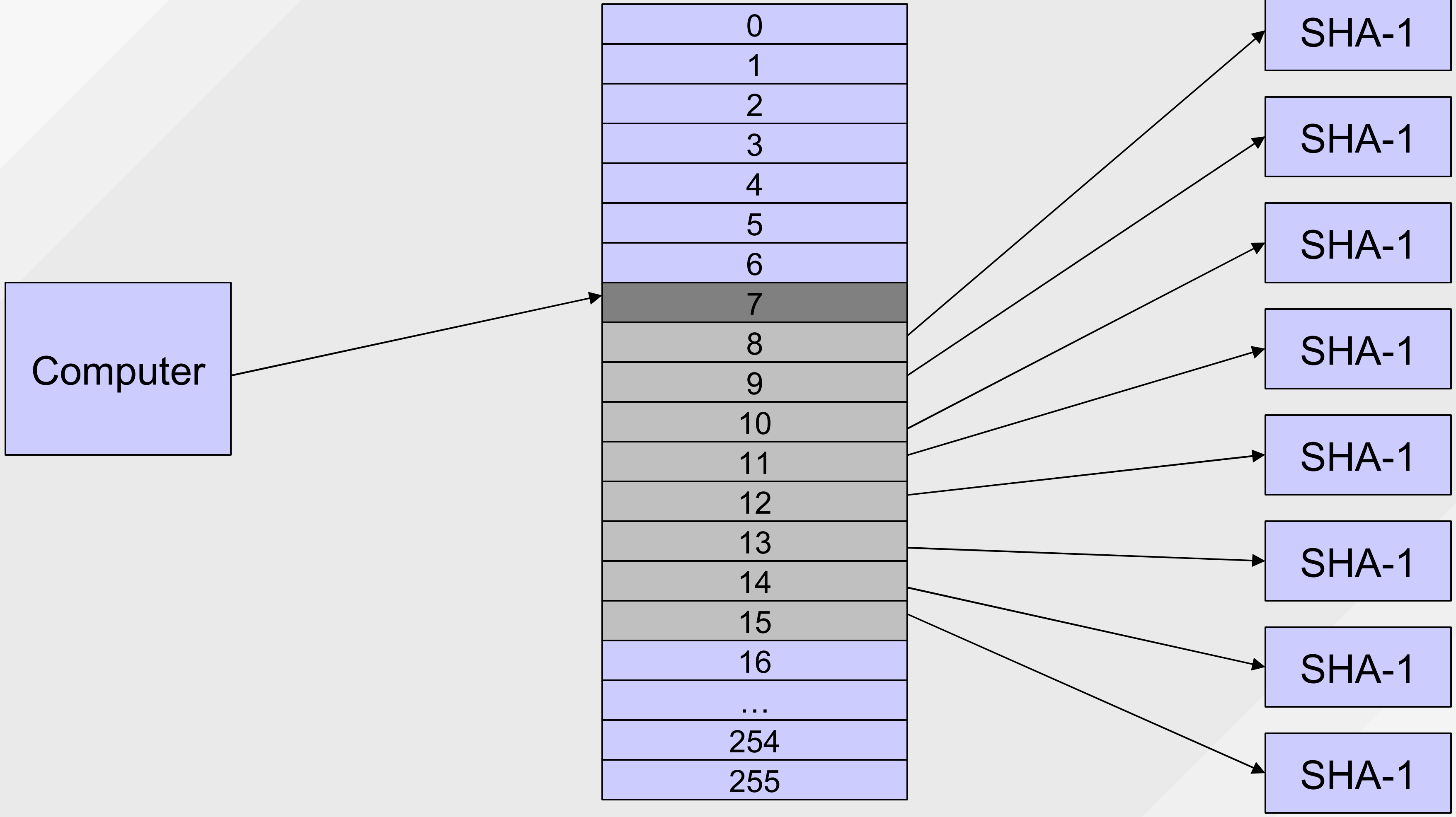
FPGA coWPAtty



FPGA coWPAtty



FPGA coWPAtty



Performance Comparison

PC		FPGA	
Cowpatty		**Cowpatty**	
800MHz P3	~25/sec	LX25	~430/sec
3.6GHz P4	~60/sec	15 Cluster	~6,500/sec
AMD Opteron	~70/sec	FX60	~1,000/sec
2.16GHz IntelDuo	~70/sec		
Aircrack			
3.6GHz P4	~100/sec		

Results

- Decided to compute hash tables for a 1,000,000 passphrase wordlist for the top 1,000 SSIDs

“That million word list that I fed you incorporated a 430,000 word list from Mark Burnett and Kevin Mitnick (of all people) and was made up of actual harvested passwords acquired through some google hacking. They are passwords that people have actually used. I padded it out to 1 million by adding things like websters dictionary, and other such lists, and then stripped the short word (<8 chars.) out of it.”

OpenCiphers

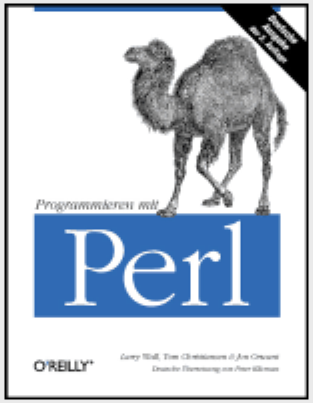
Results

- Took RenderMan 1 month to compute on his cluster
- Found out that his wordlist had return characters at the end of every line
- (after computing for a month)
- He sent me an email asking for help
- A 15 card cluster did it in 2 days ;-)

FPGA coWPAtty

+

+



= ?

Demo

Mac OS-X coWPAtty???

- /System/Library/PrivateFrameworks/Apple80211.framework/Versions/Current/Resources/airport

```
airport AirPort v.427.2 (427.2.0)

Supported arguments:
<snip a whole bunch of semi-normal iwconfig-like features>
-P<arg> --psk=<arg>    Create PSK from specified passphrase
                       and SSID.
The following additional arguments must be specified with this
   command:
--ssid=<arg>           Specify SSID when
```

OpenCiphers

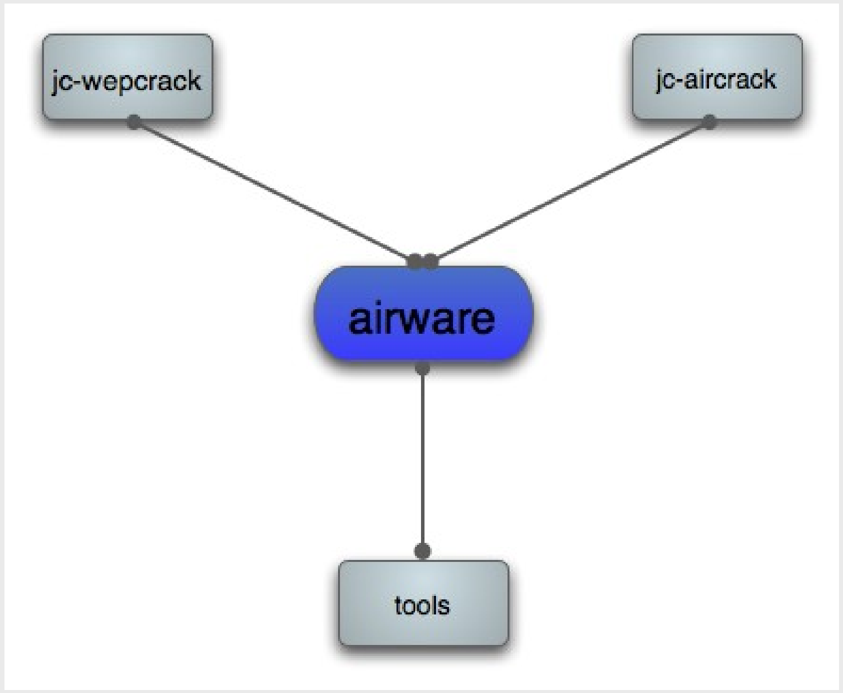
ghetto_pmk.pl

```
#!/usr/bin/perl

open(INFILE,"dictionary.txt");
my $start = time;
my $count = 0;
foreach (<INFILE>) {
        chop($_);
        $cmd = "airport --psk=$_ --ssid=linksys >> pmks.txt";
        system $cmd;
        $count++;
}

$elapsed = time - $start;
$perform = $count / $elapsed;
print "$count passphrases tested in $elapsed seconds: ";
print "$perform passphrases/second\n";

beetles-computer:~/Downloads beetle$ ./ghetto_pmk.pl
2253 passphrases tested in 217 seconds: 10.3824884792627 passphrases/second
```

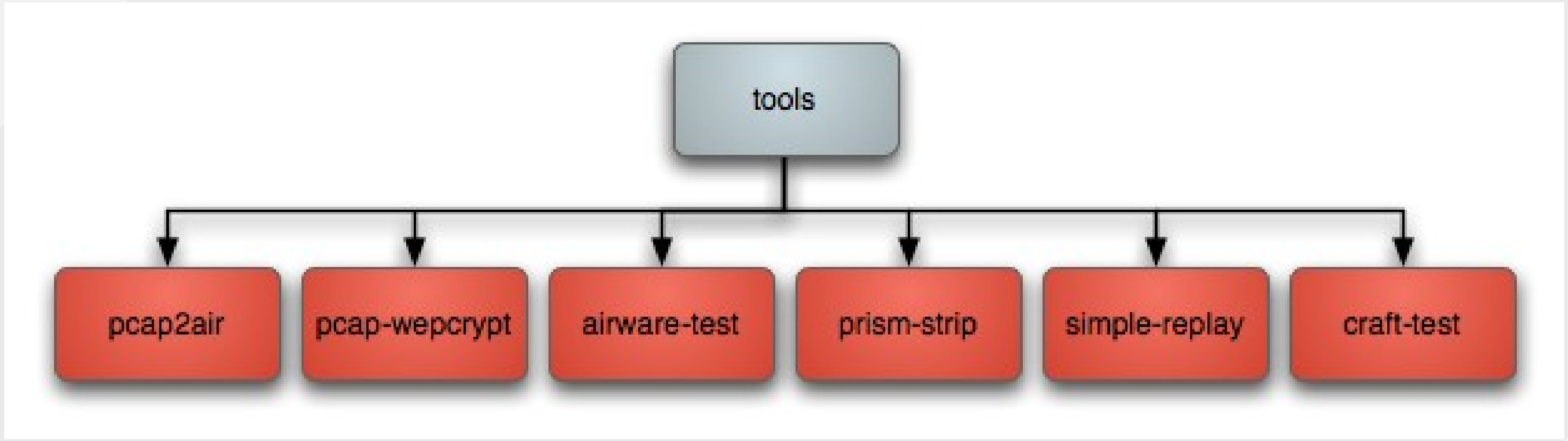
Airbase

OpenCiphers

Airbase

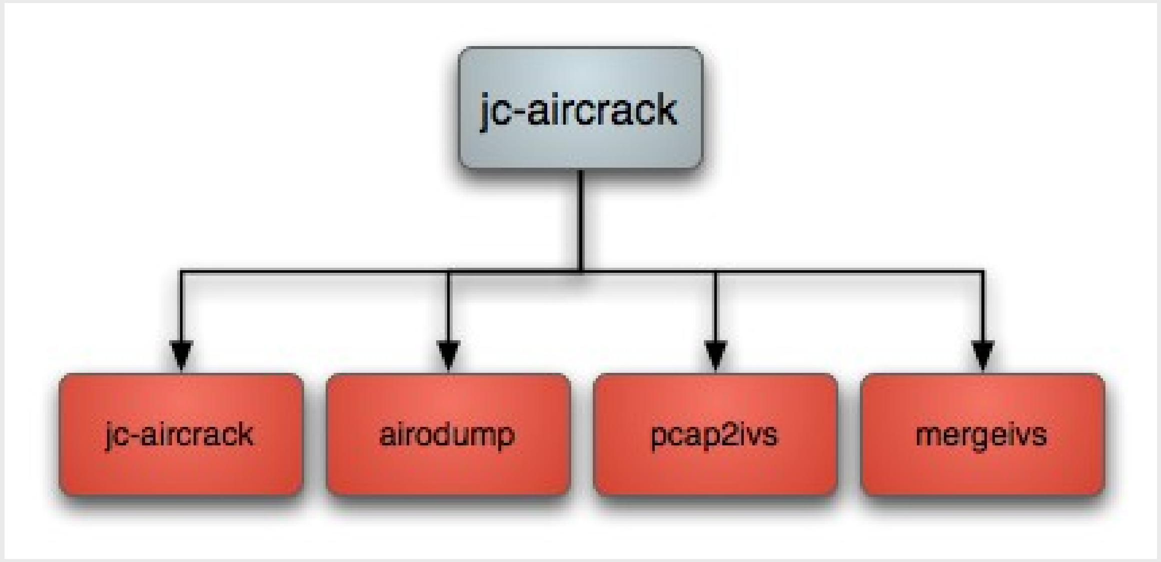


Airbase

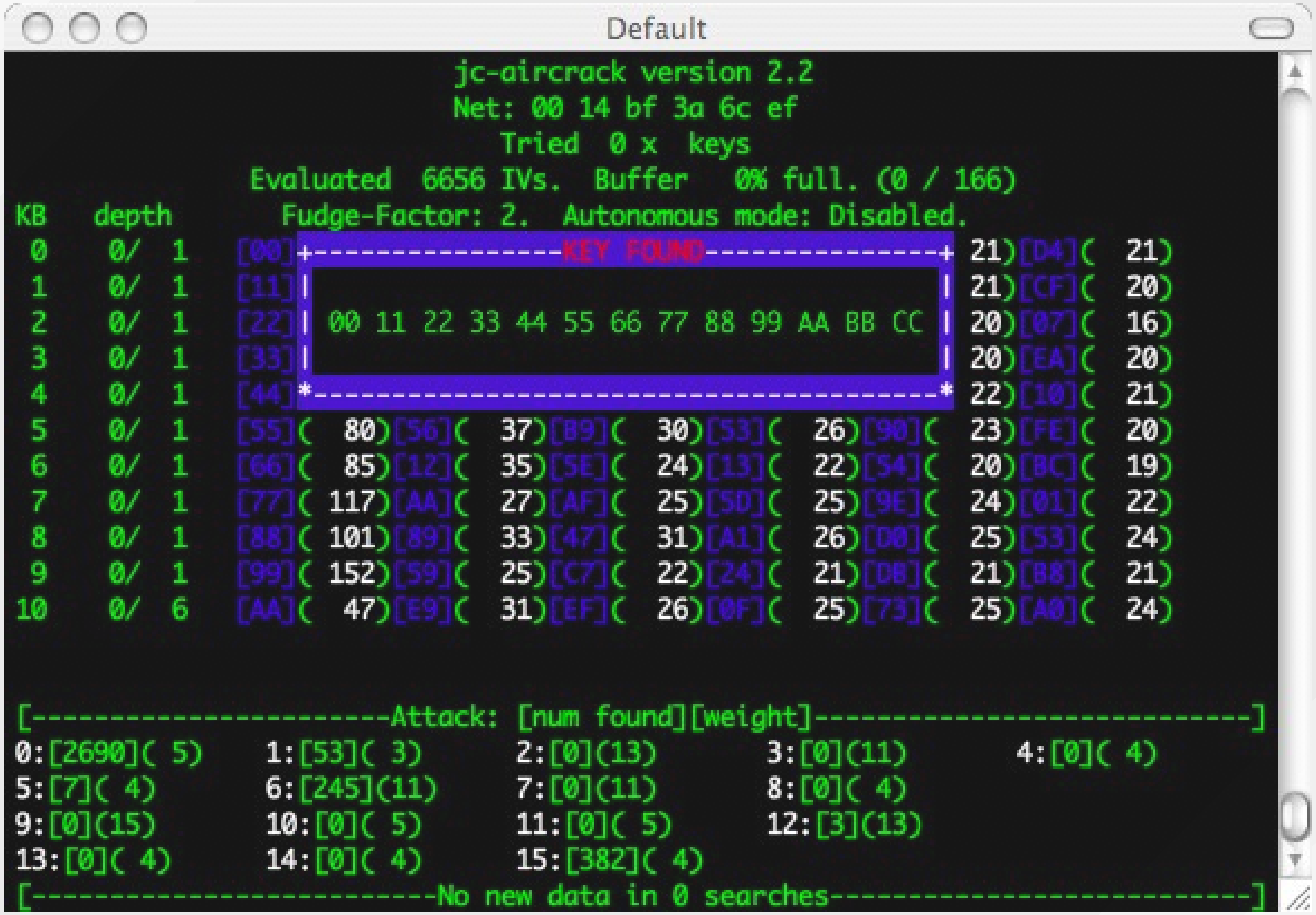

tools
pcap2air
pcap-wepcrypt
airware-test
prism-strip
simple-replay
craft-test

OpenCiphers

jc-aircrack

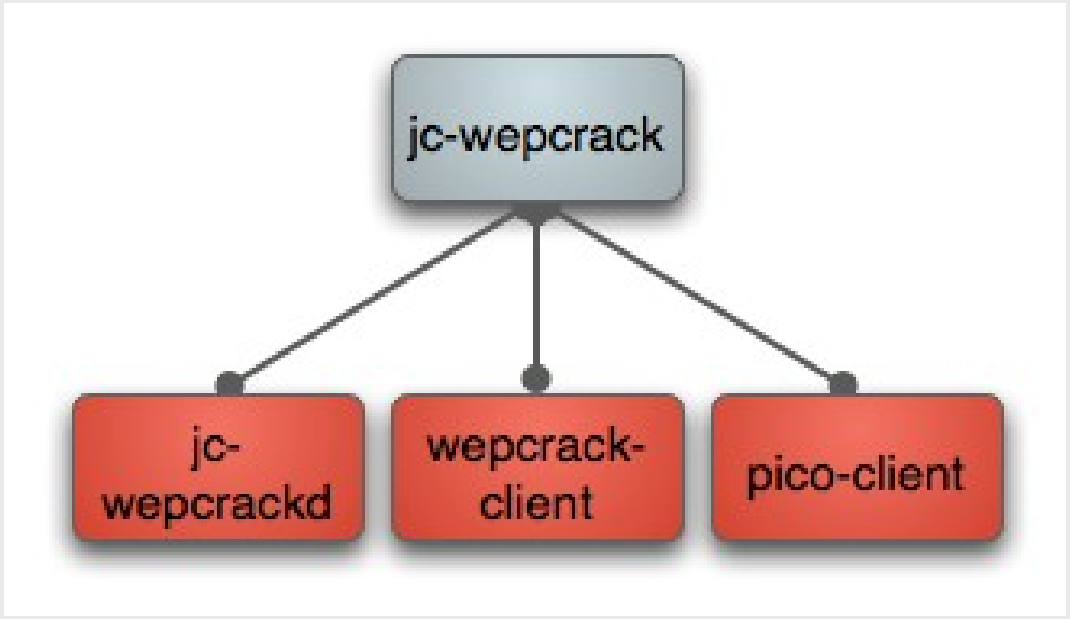


jc-aircrack

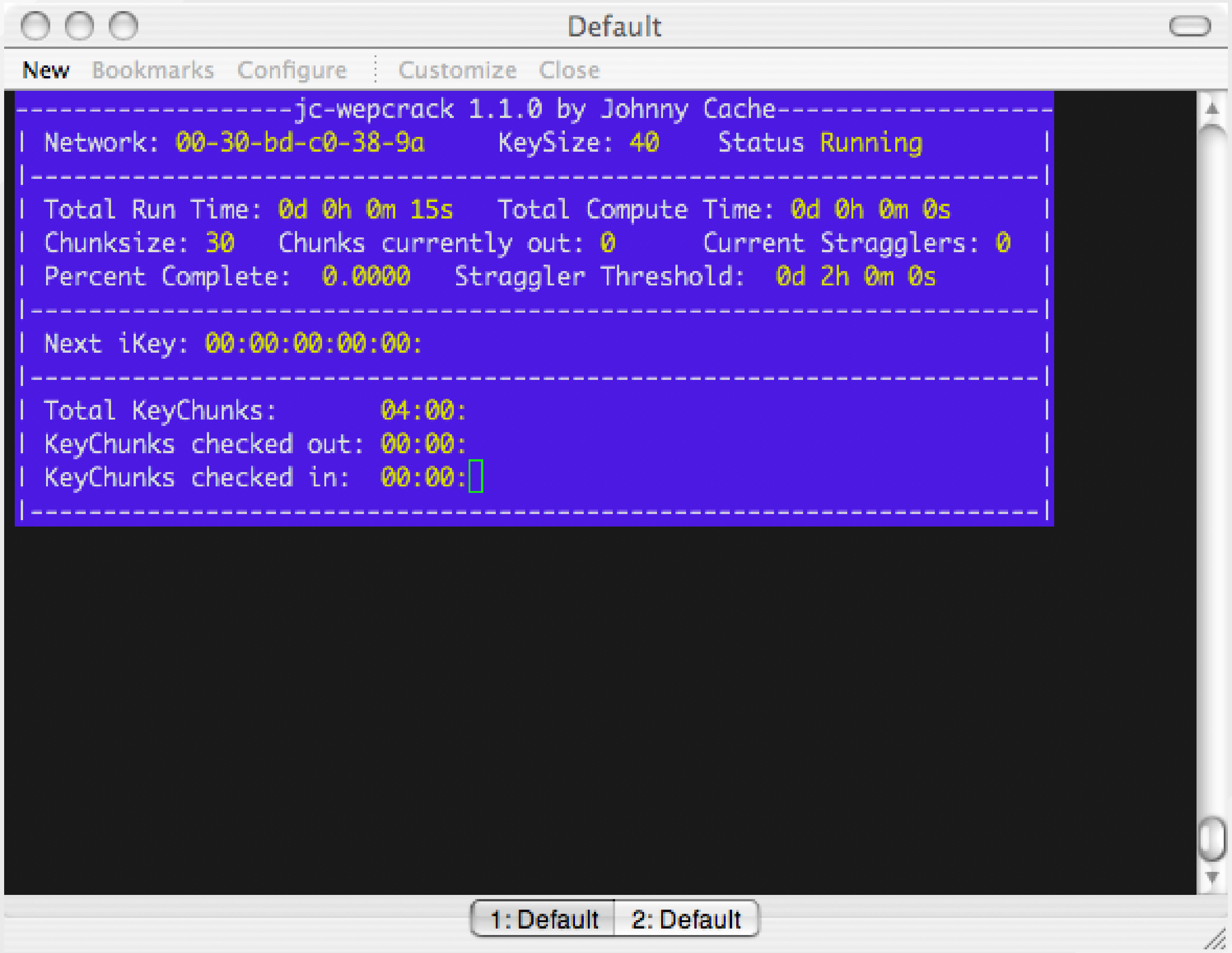

Default

```
                              jc-aircrack version 2.2
                             Net: 00 14 bf 3a 6c ef
                                Tried  0 x  keys
                Evaluated  6656 IVs.  Buffer   0% full. (0 / 166)
KB   depth        Fudge-Factor: 2.  Autonomous mode: Disabled.
 0    0/  1     [00]+----------------KEY FOUND---------------+ 21)[D4](  21)
 1    0/  1     [11]|                                         | 21)[CF](  20)
 2    0/  1     [22]| 00 11 22 33 44 55 66 77 88 99 AA BB CC | 20)[07](  16)
 3    0/  1     [33]|                                         | 20)[EA](  20)
 4    0/  1     [44]*-----------------------------------------* 22)[10](  21)
 5    0/  1     [55](  80)[56](  37)[B9](  30)[53](  26)[90](  23)[FE](  20)
 6    0/  1     [66](  85)[12](  35)[5E](  24)[13](  22)[54](  20)[BC](  19)
 7    0/  1     [77]( 117)[AA](  27)[AF](  25)[5D](  25)[9E](  24)[01](  22)
 8    0/  1     [88]( 101)[89](  33)[47](  31)[A1](  26)[D0](  25)[53](  24)
 9    0/  1     [99]( 152)[59](  25)[C7](  22)[24](  21)[DB](  21)[B8](  21)
10    0/  6     [AA](  47)[E9](  31)[EF](  26)[0F](  25)[73](  25)[A0](  24)


[-----------------------Attack: [num found][weight]----------------------------]
0:[2690]( 5)     1:[53]( 3)       2:[0](13)        3:[0](11)        4:[0]( 4)
5:[7]( 4)        6:[245](11)      7:[0](11)        8:[0]( 4)
9:[0](15)        10:[0]( 5)       11:[0]( 5)       12:[3](13)
13:[0]( 4)       14:[0]( 4)       15:[382]( 4)
[------------------------No new data in 0 searches----------------------------]
```

jc-wepcrack

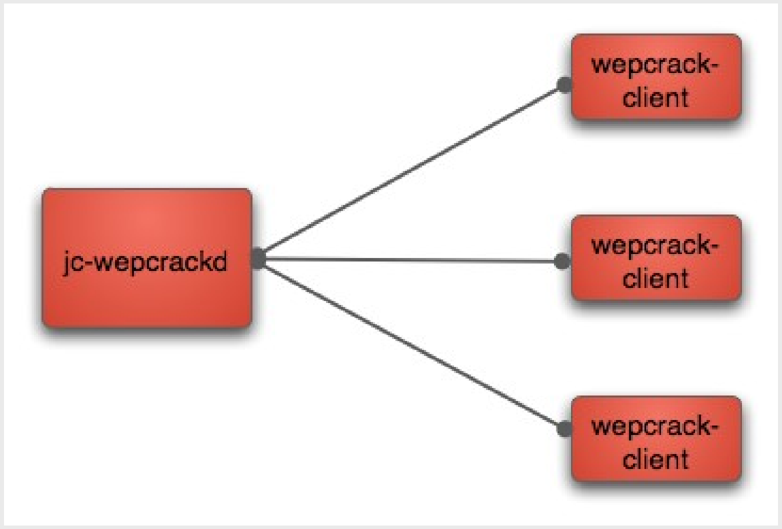


jc-wepcrack

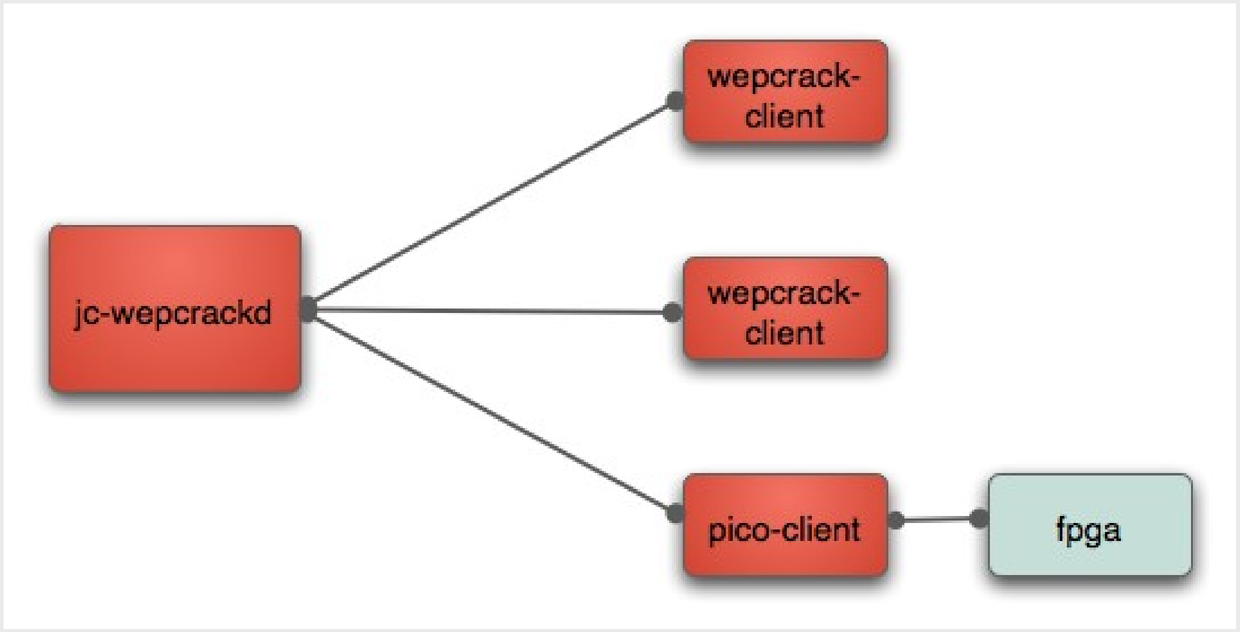


```
-------------------jc-wepcrack 1.1.0 by Johnny Cache--------------------
| Network: 00-30-bd-c0-38-9a      KeySize: 40     Status Running        |
|-----------------------------------------------------------------------|
| Total Run Time: 0d 0h 0m 15s    Total Compute Time: 0d 0h 0m 0s       |
| Chunksize: 30   Chunks currently out: 0        Current Stragglers: 0  |
| Percent Complete:  0.0000    Straggler Threshold:  0d 2h 0m 0s        |
|-----------------------------------------------------------------------|
| Next iKey: 00:00:00:00:00:                                            |
|-----------------------------------------------------------------------|
| Total KeyChunks:        04:00:                                        |
| KeyChunks checked out: 00:00:                                         |
| KeyChunks checked in:  00:00:                                         |
|-----------------------------------------------------------------------|
```

jc-wepcrack, no pico

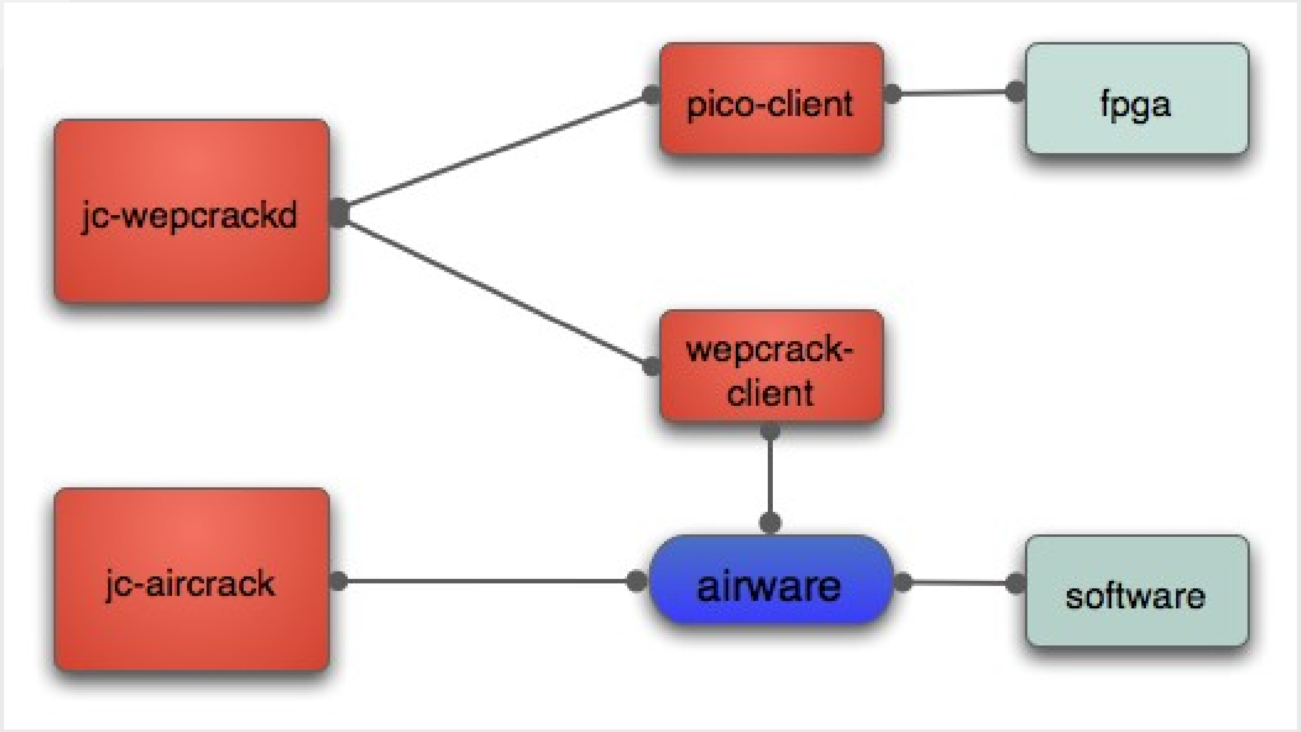


Accelerating brute forcing

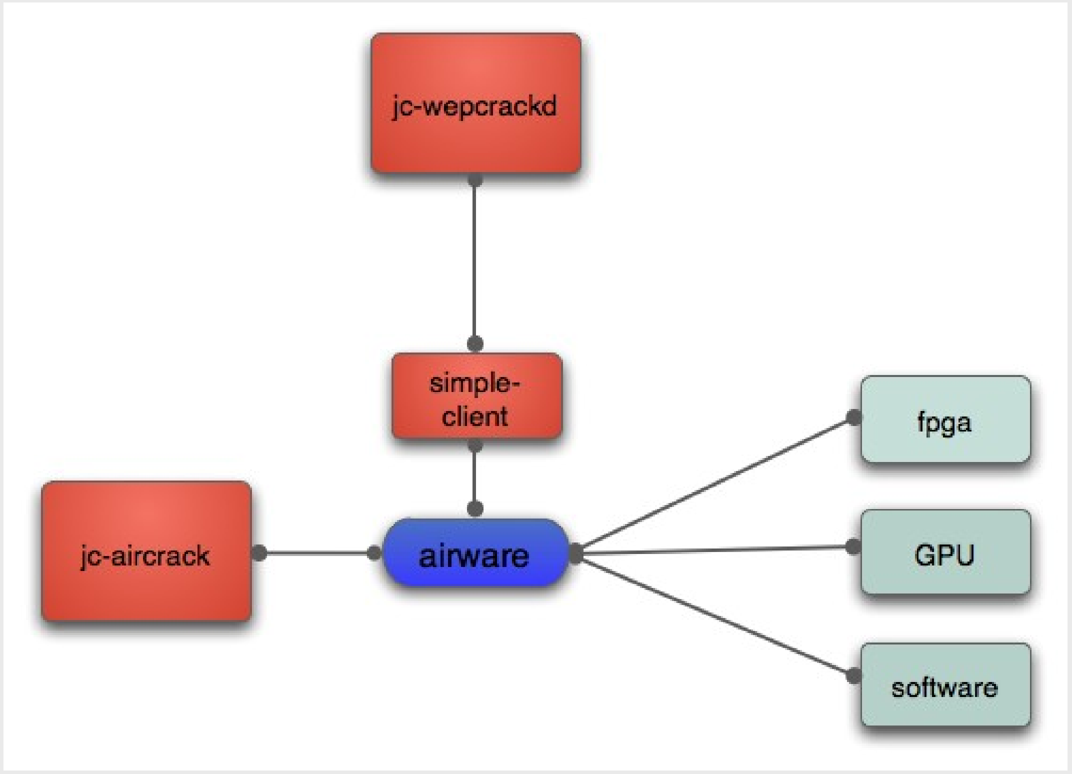


OpenCiphers

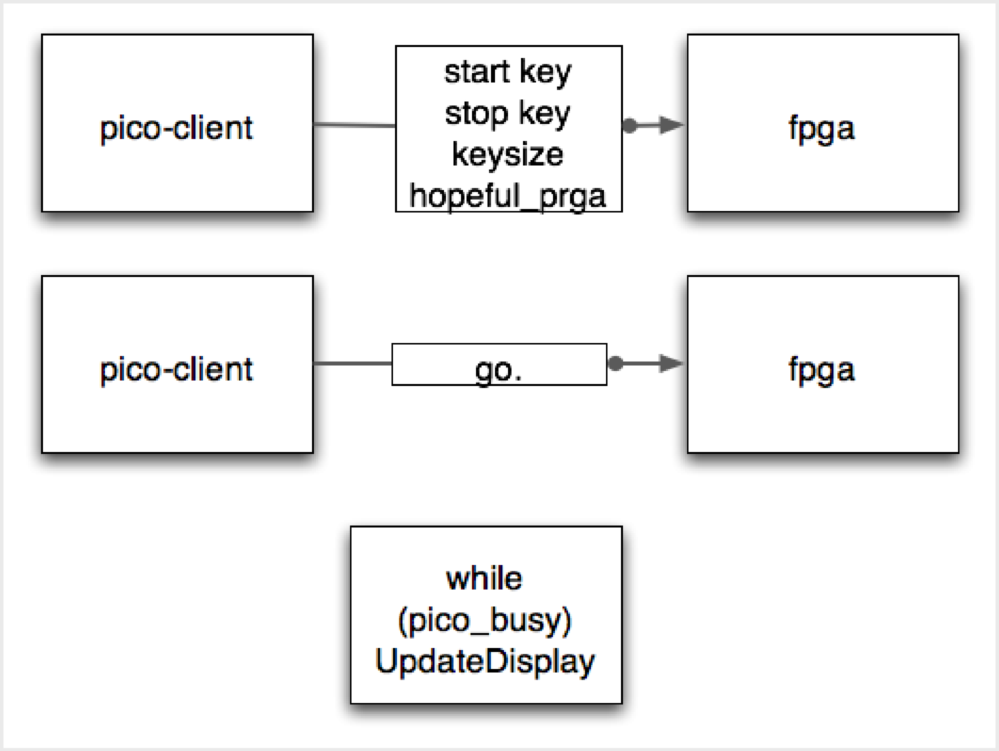
Current arch



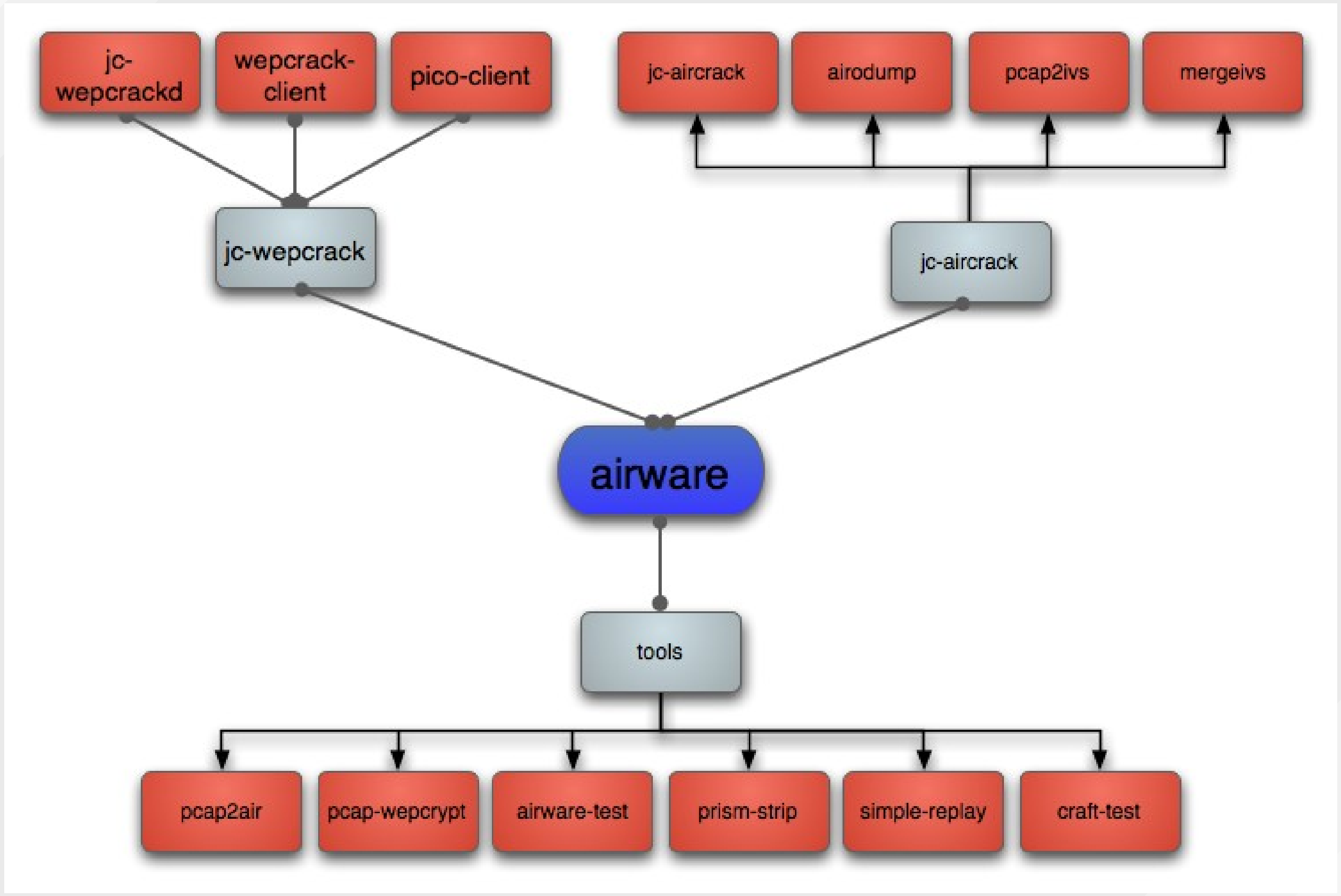
Future arch


jc-wepcrackd
simple-client
jc-aircrack
airware
fpga
GPU
software

Pico client details



Airbase


jc-wepcrackd
wepcrack-client
pico-client
jc-aircrack
airodump
pcap2ivs
mergeivs
jc-wepcrack
jc-aircrack
airware
tools
pcap2air
pcap-wepcrypt
airware-test
prism-strip
simple-replay
craft-test

pico-wepcrack

- FPGA Core
  - Uses 32/48 custom RC4 cores
  - Uses BlockRAM for S-Boxes
  - Will try every key between a start and end

pico-wepcrack

- RC4:

```
for(i = 0; i < 256; i++)                     // Initialization
  S[i] = i;

for(i = j = 0; i < 256; i++) {               // KSA
  j += S[i] + K[i];
  Swap(S[i], S[j]);
}

for(i = 1, j = 0; ; i++) {                   // PRGA
  j += S[i];
  Swap(S[i], S[j]);
  PRGA[i - 1] = S[S[i] + S[j]];
}
```

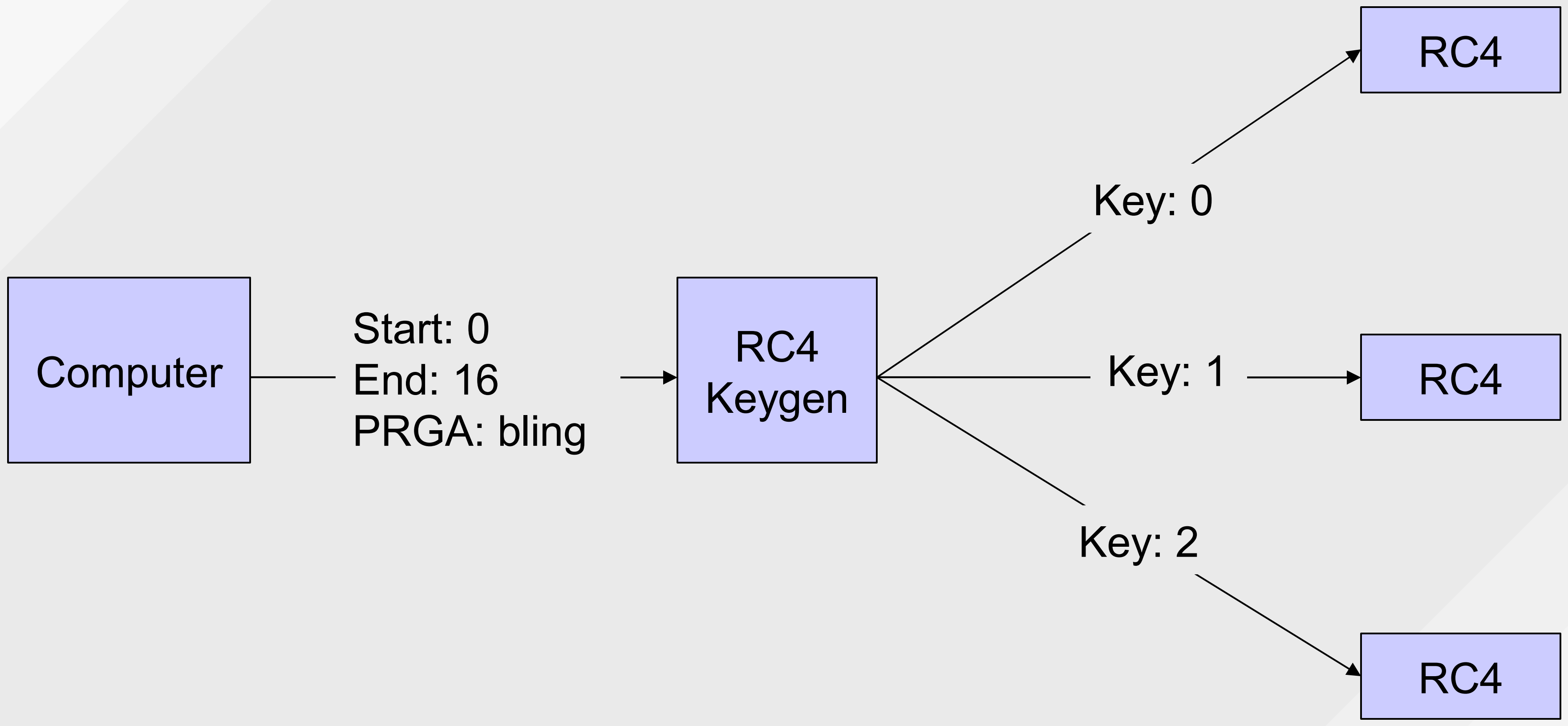
OpenCiphers

pico-wepcrack

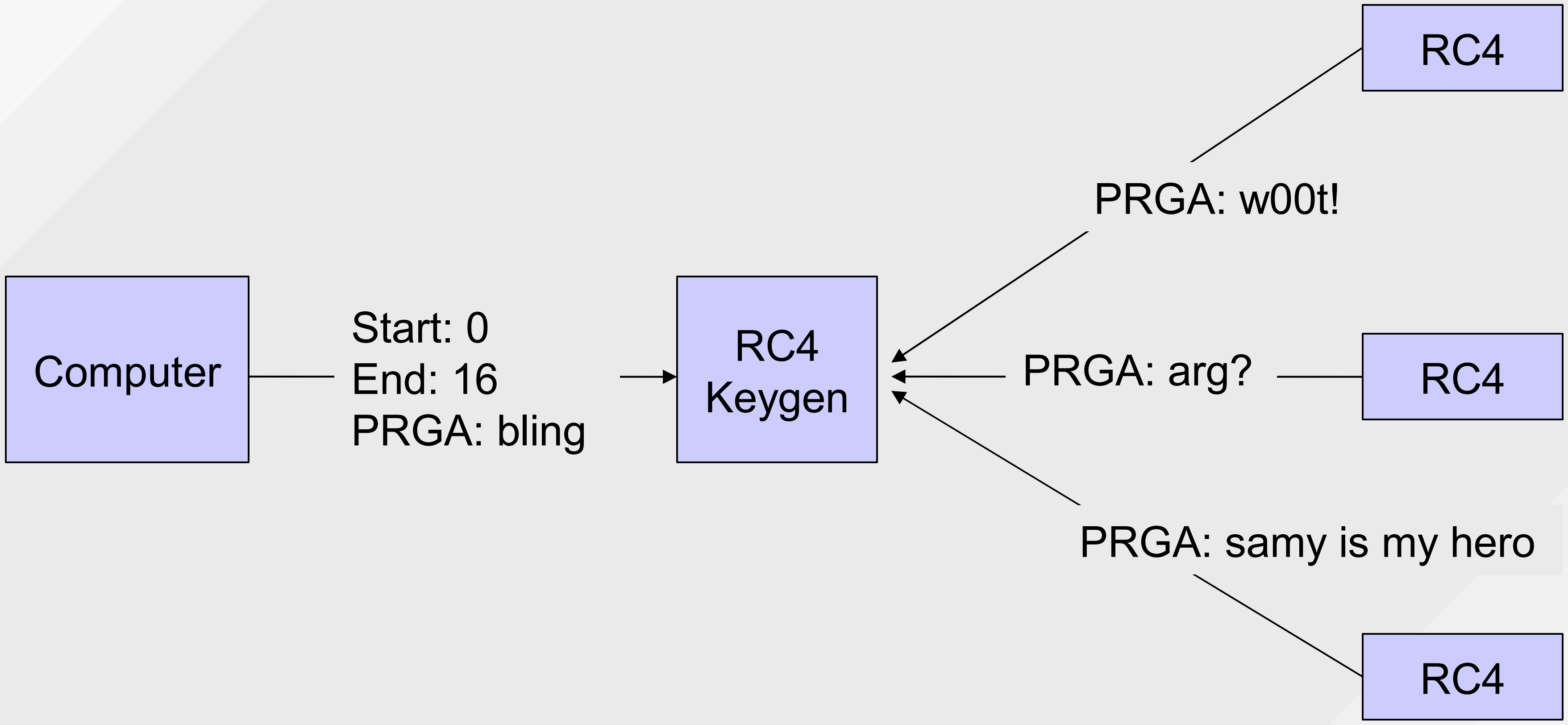
- RC4:

```
for(i = 0; i < 256; i++)                          // Initialization
  S[i] = i;

for(i = j = 0; i < 256; i++) {                    // KSA
  j += S[i] + K[i];                               // K is input
  Swap(S[i], S[j]);
}

for(i = 1, j = 0; ; i++) {                        // PRGA
  j += S[i];
  Swap(S[i], S[j]);
  PRGA[i - 1] = S[S[i] + S[j]];                   // PRGA is output
}
```

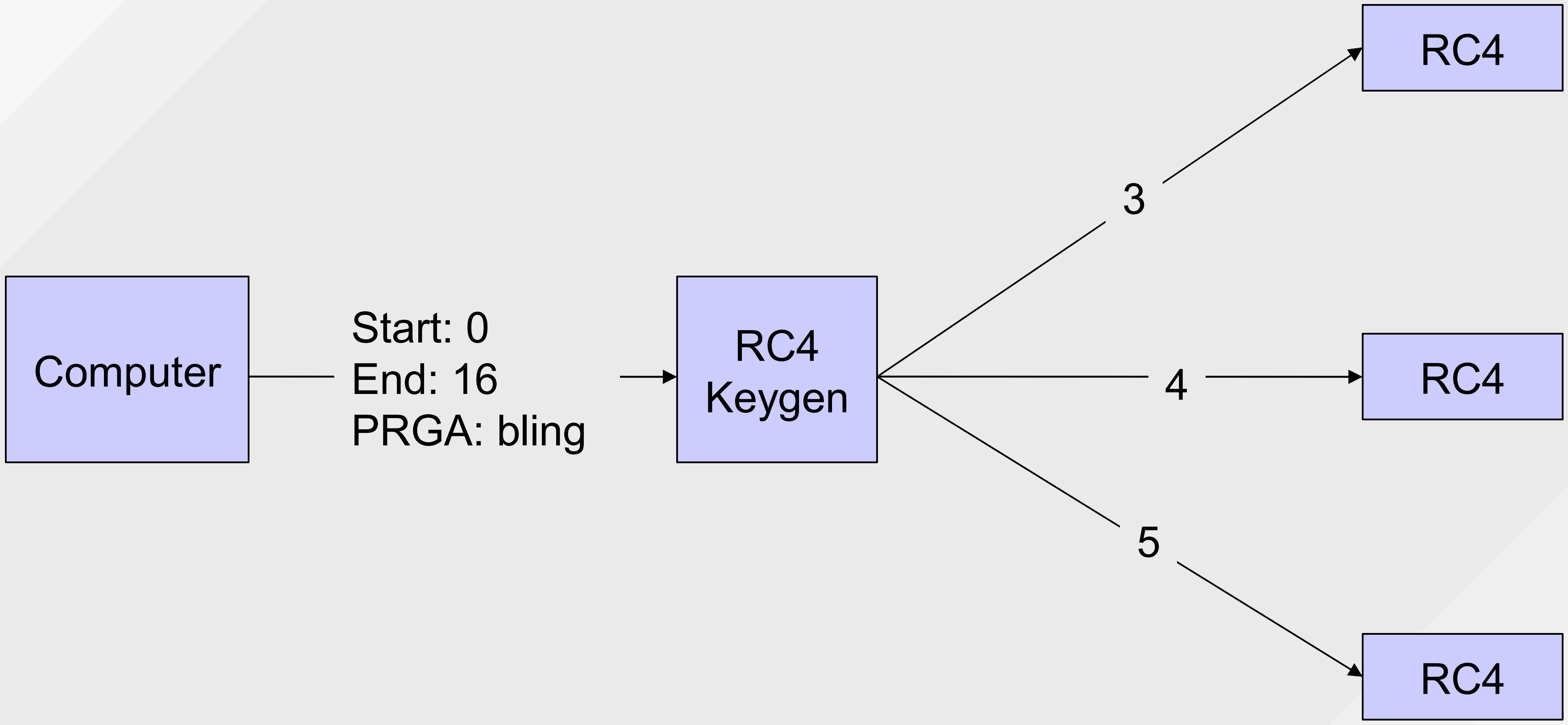
pico-wepcrack



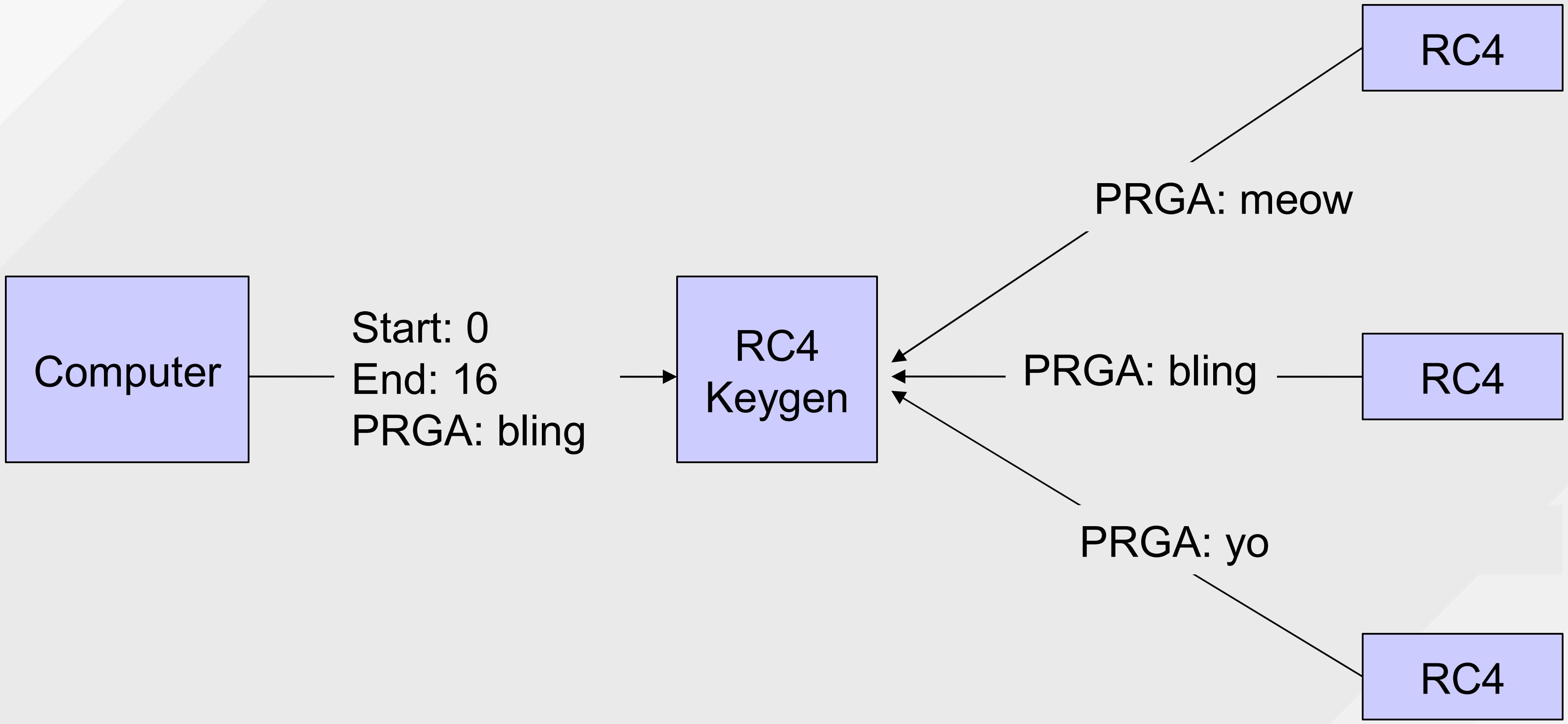
pico-wepcrack



pico-wepcrack



pico-wepcrack

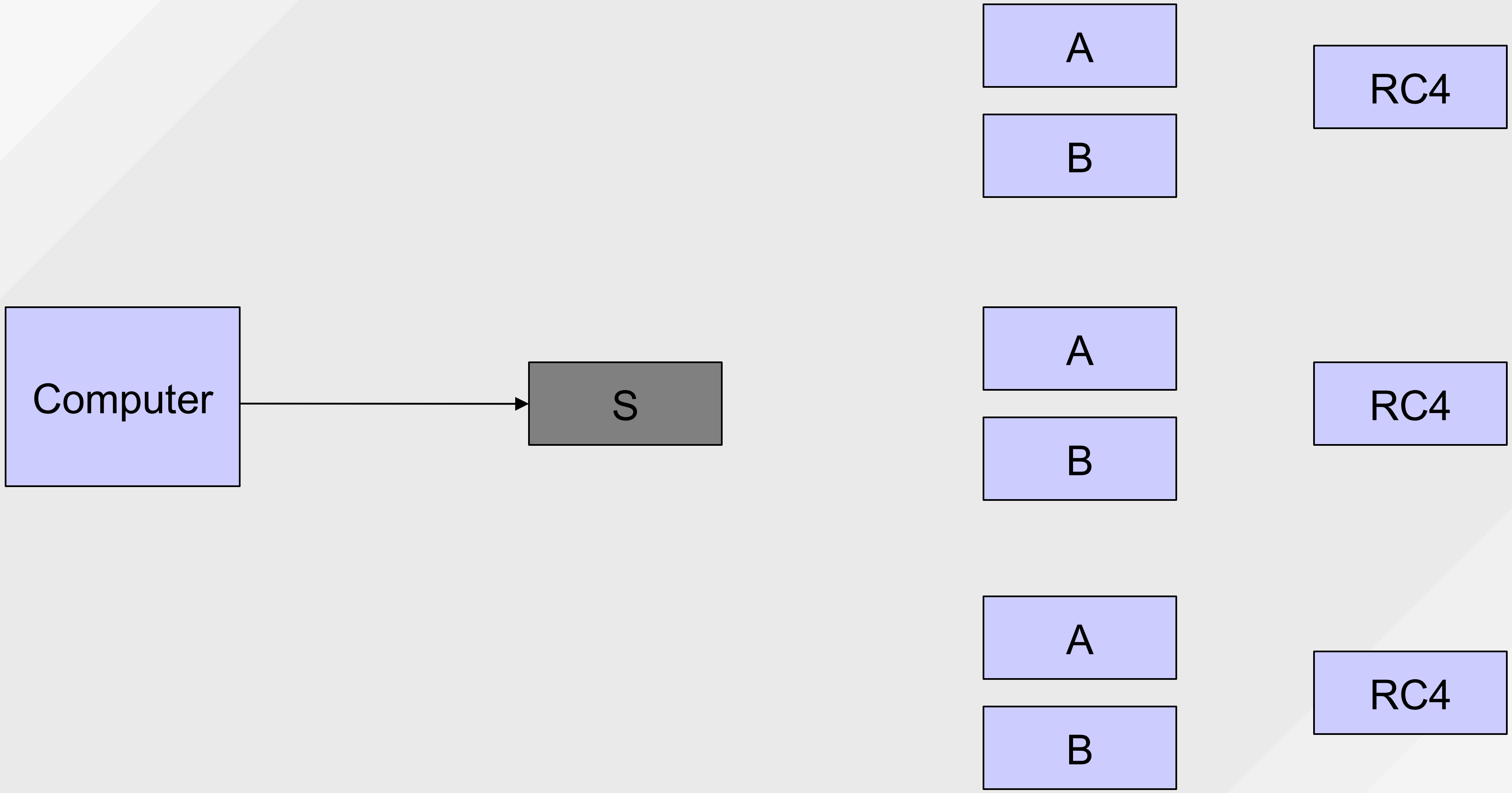


pico-wepcrack

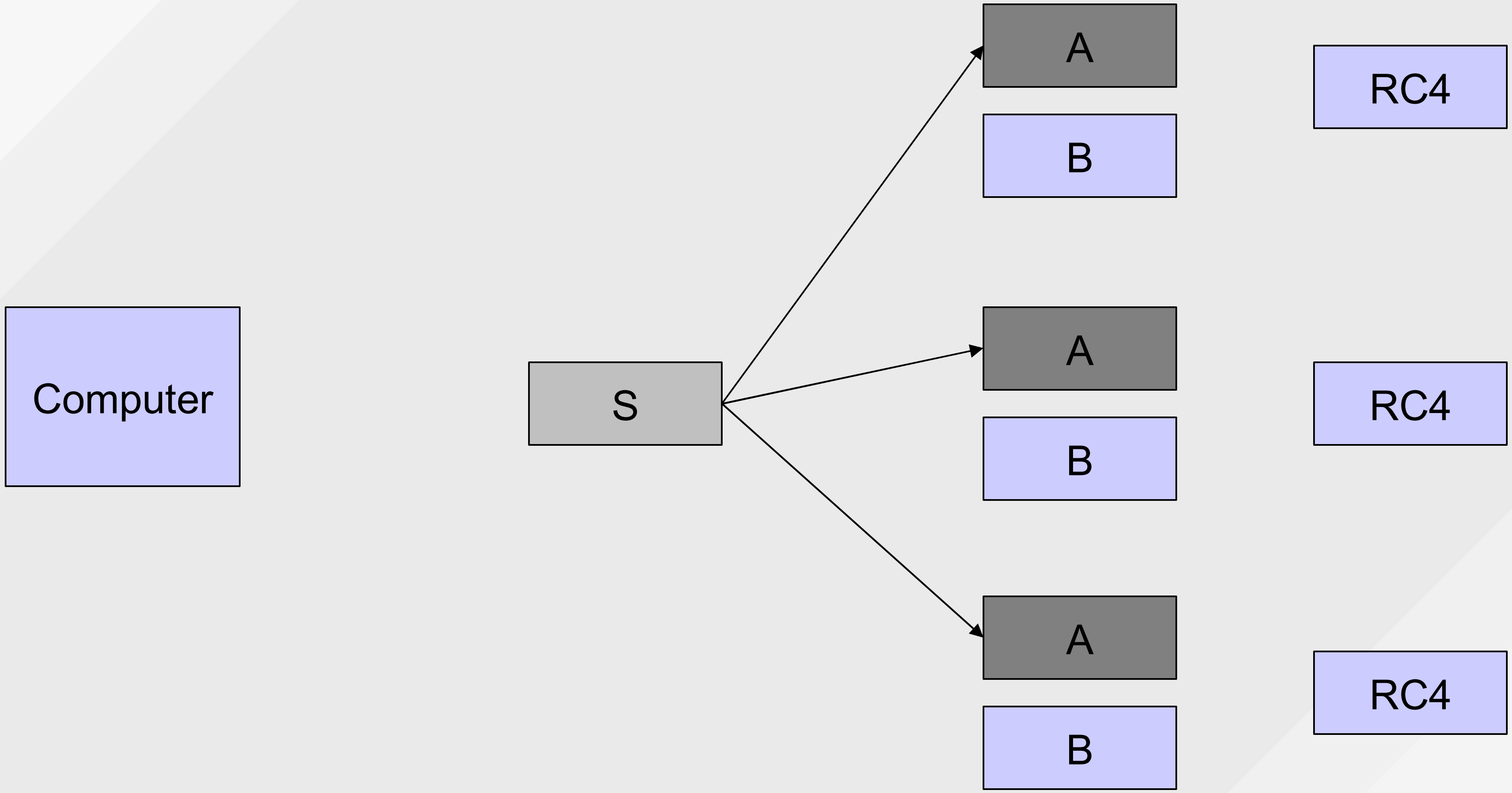
- RC4:

```
for(i = 0; i < 256; i++)                    // Initialization
   S[i] = i;                                // S-Box must be reset

for(i = j = 0; i < 256; i++) {              // KSA
   j += S[i] + K[i];
   Swap(S[i], S[j]);
}

for(i = 1, j = 0; ; i++) {                  // PRGA
   j += S[i];
   Swap(S[i], S[j]);
   PRGA[i - 1] = S[S[i] + S[j]];
}
```

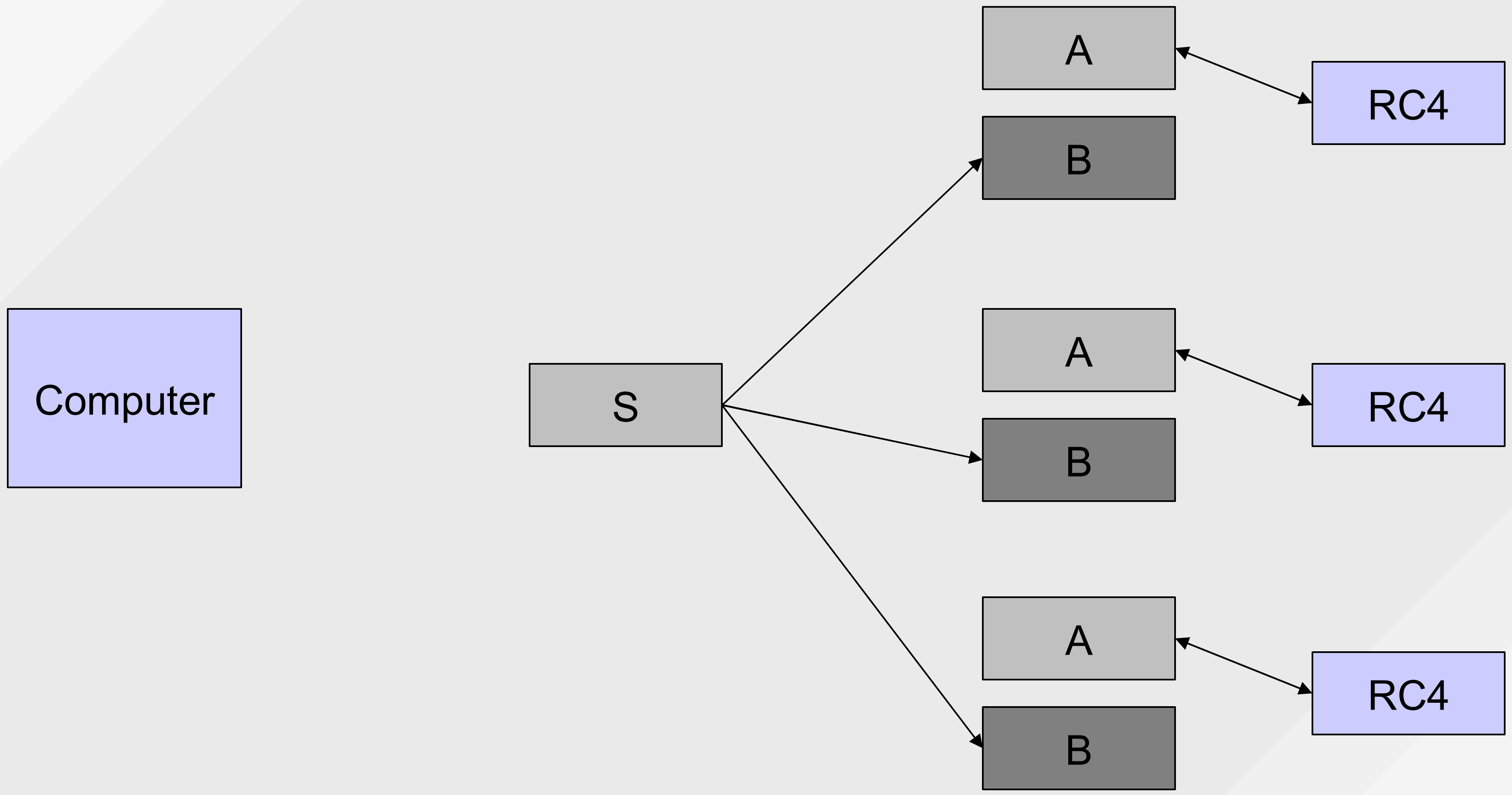
pico-wepcrack



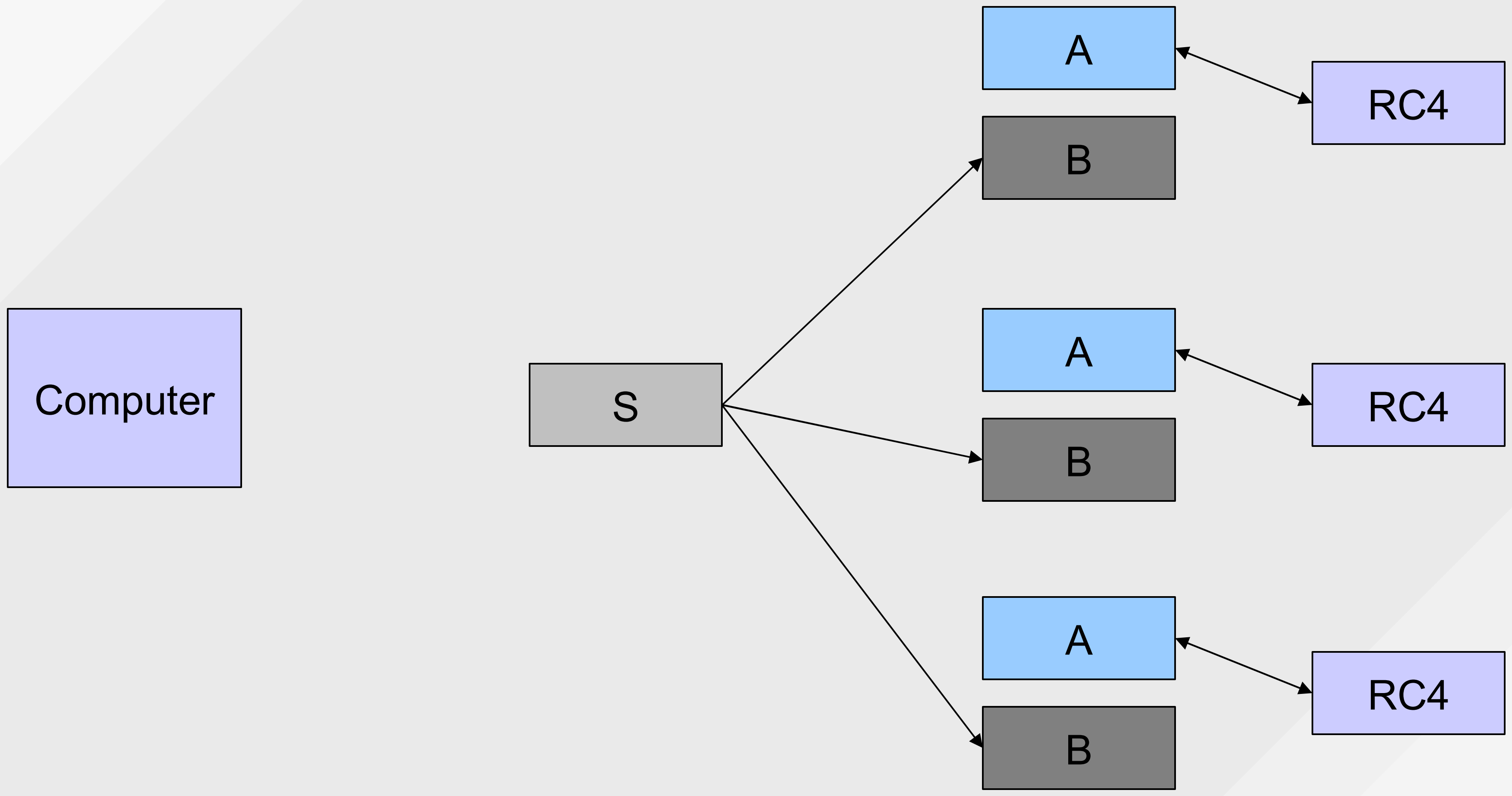
pico-wepcrack



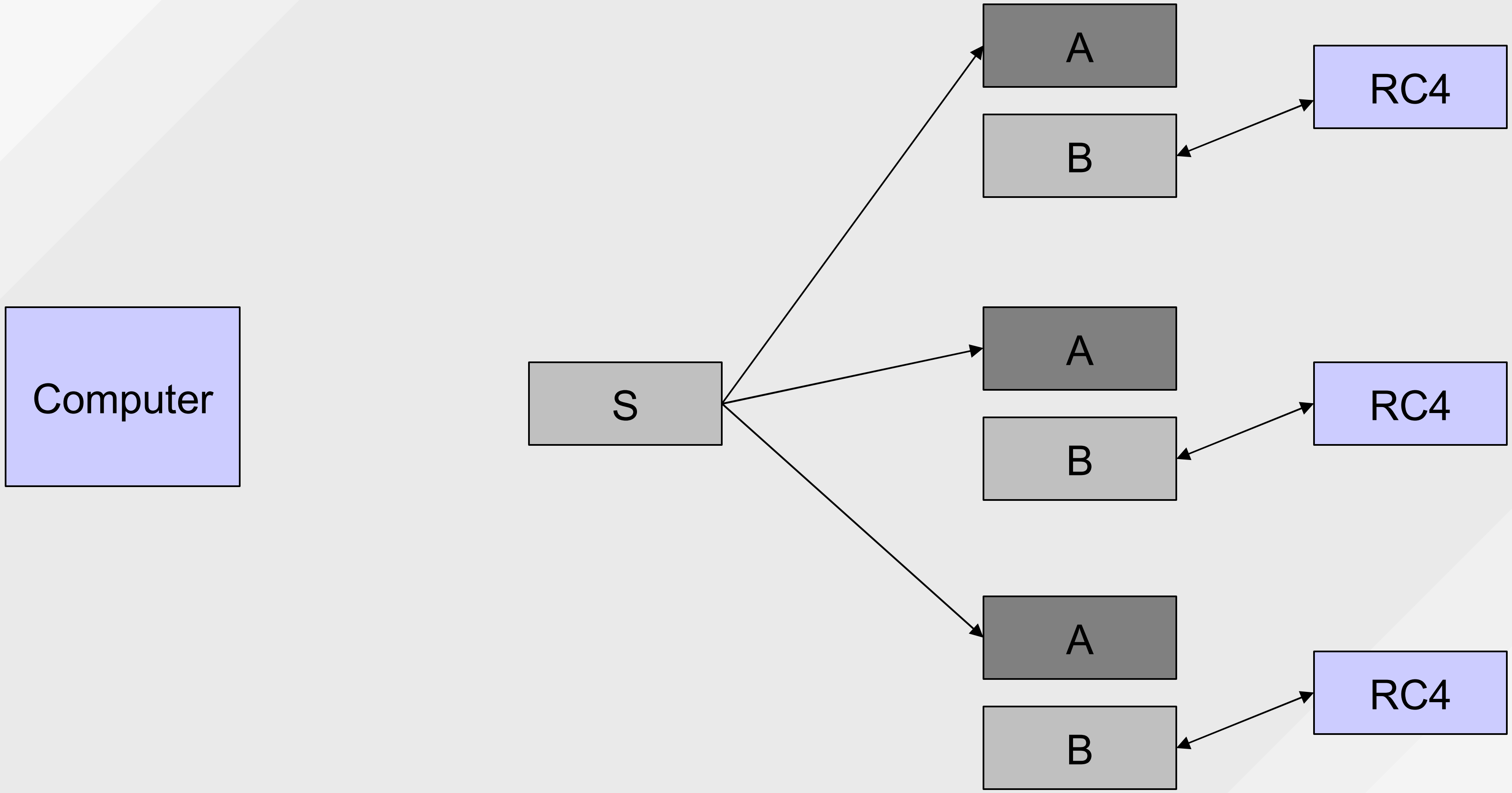
pico-wepcrack



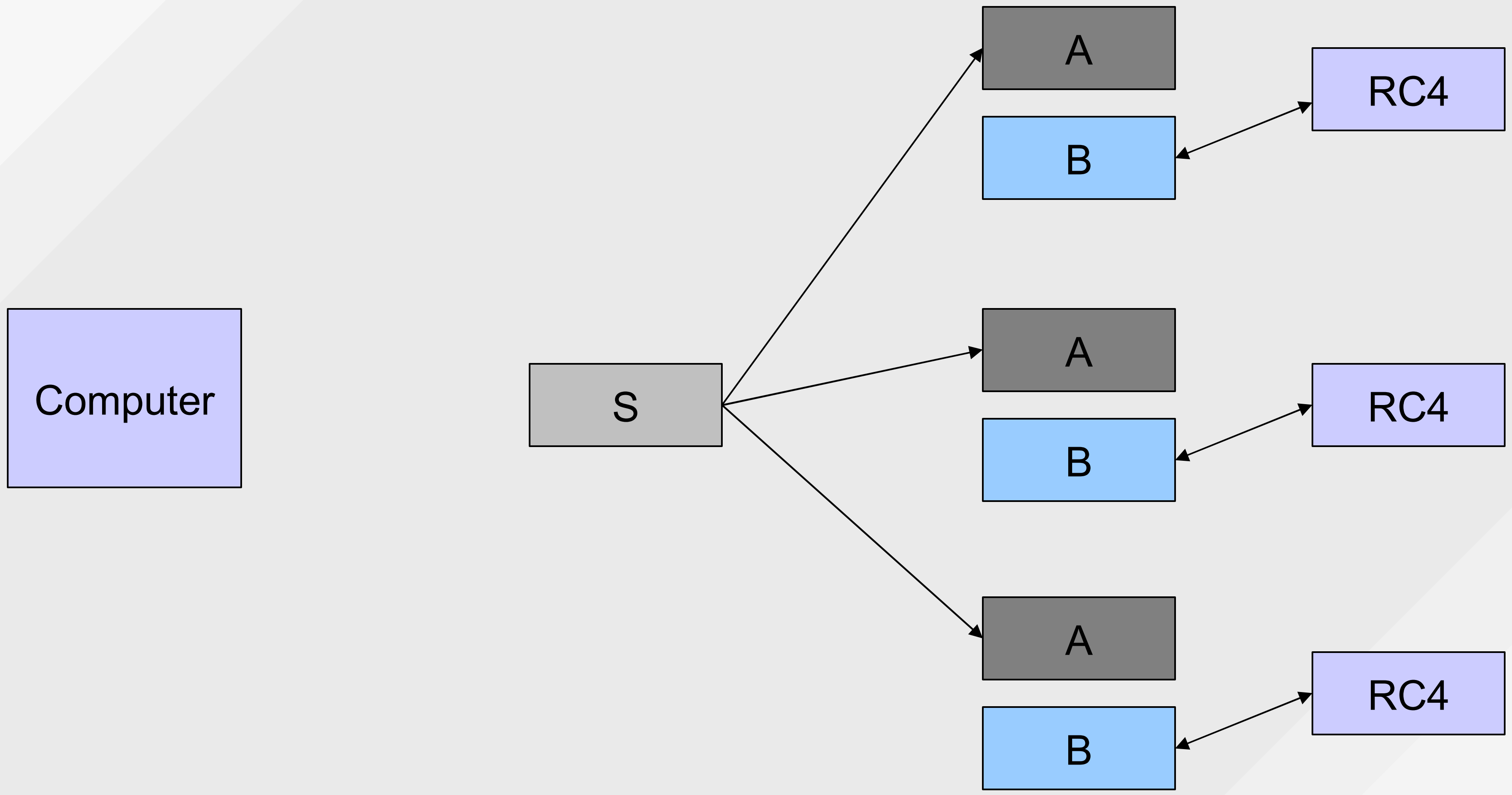
pico-wepcrack



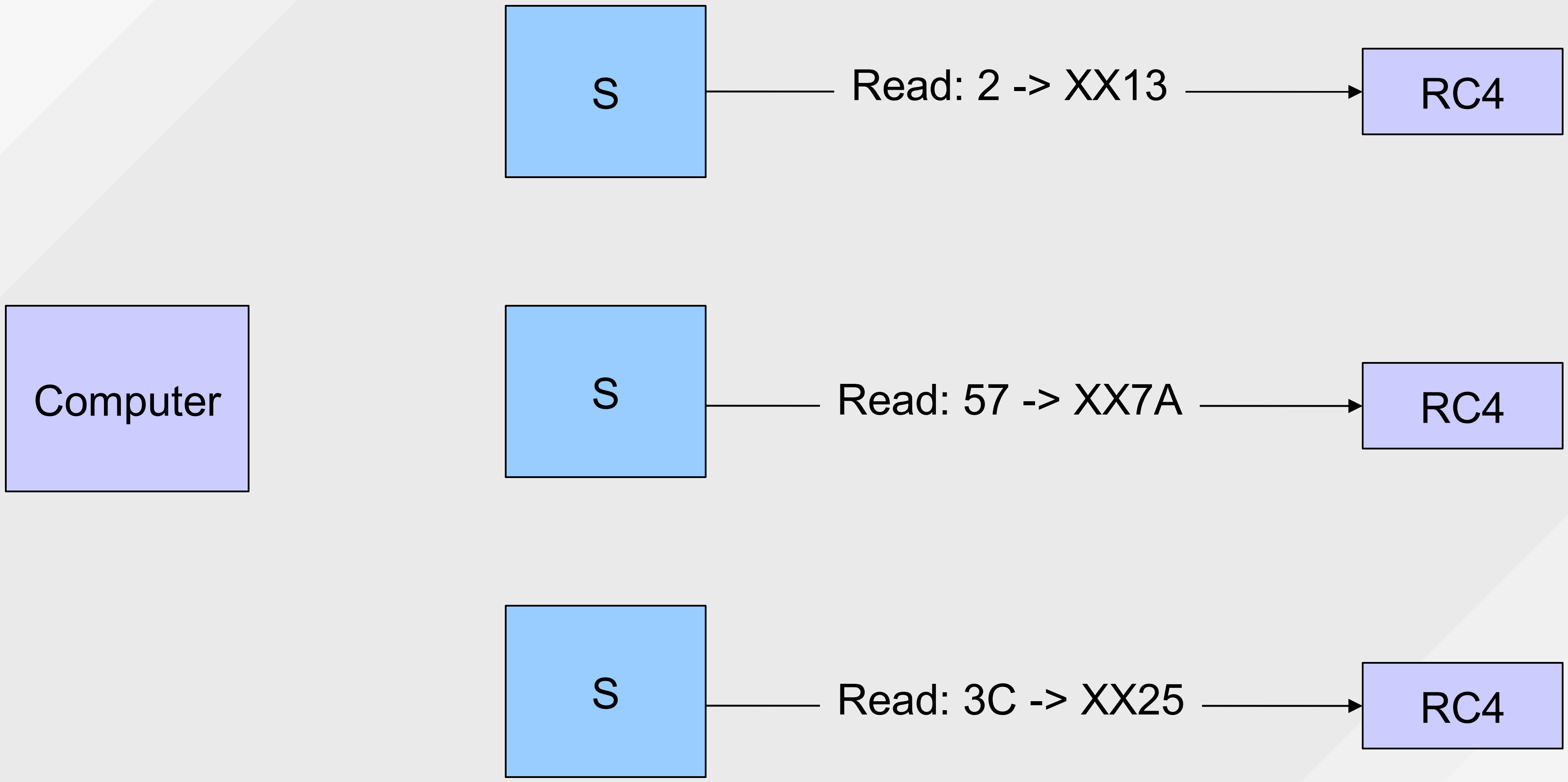
pico-wepcrack



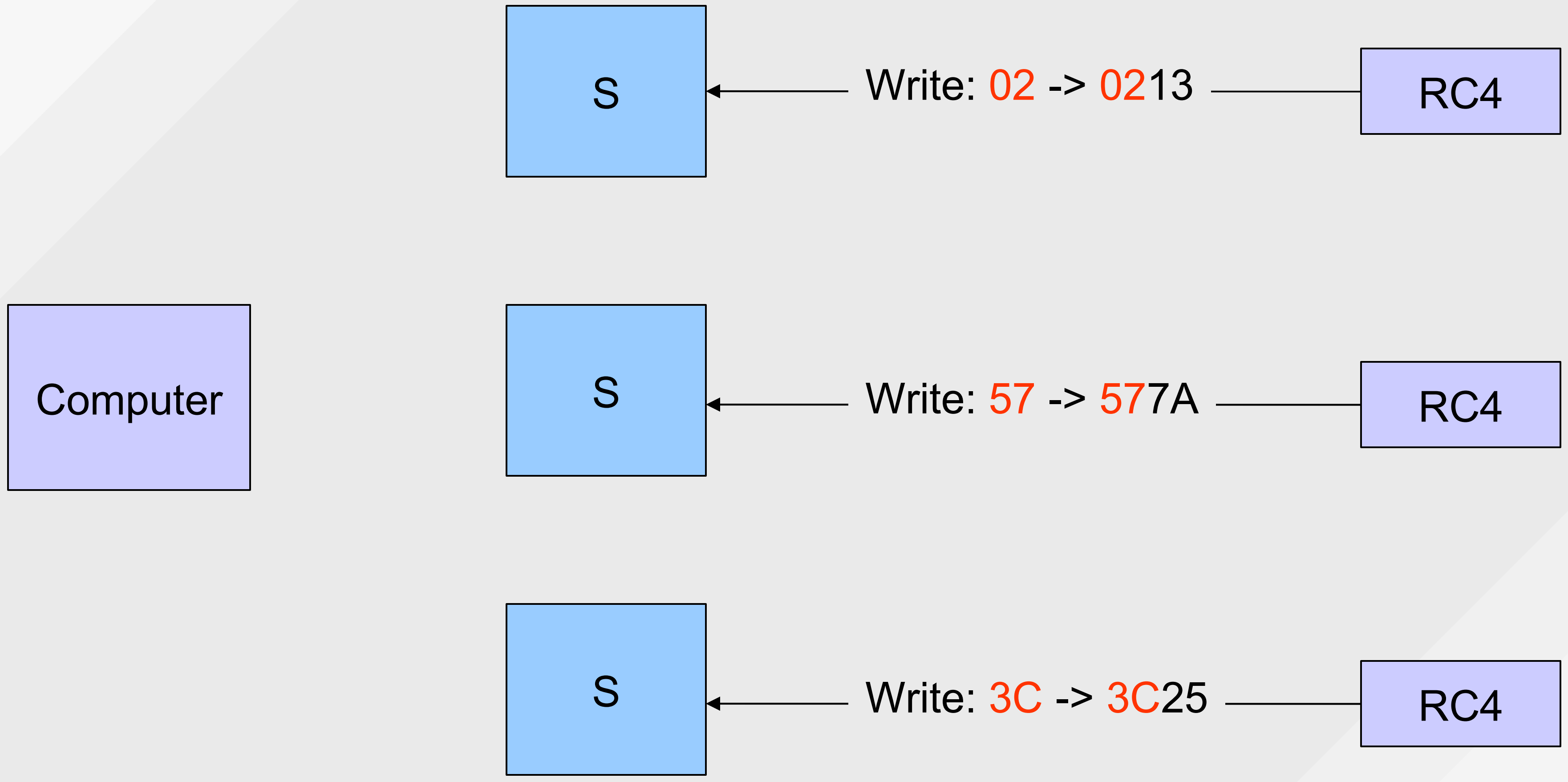
pico-wepcrack



pico-wepcrack



pico-wepcrack



OpenCiphers

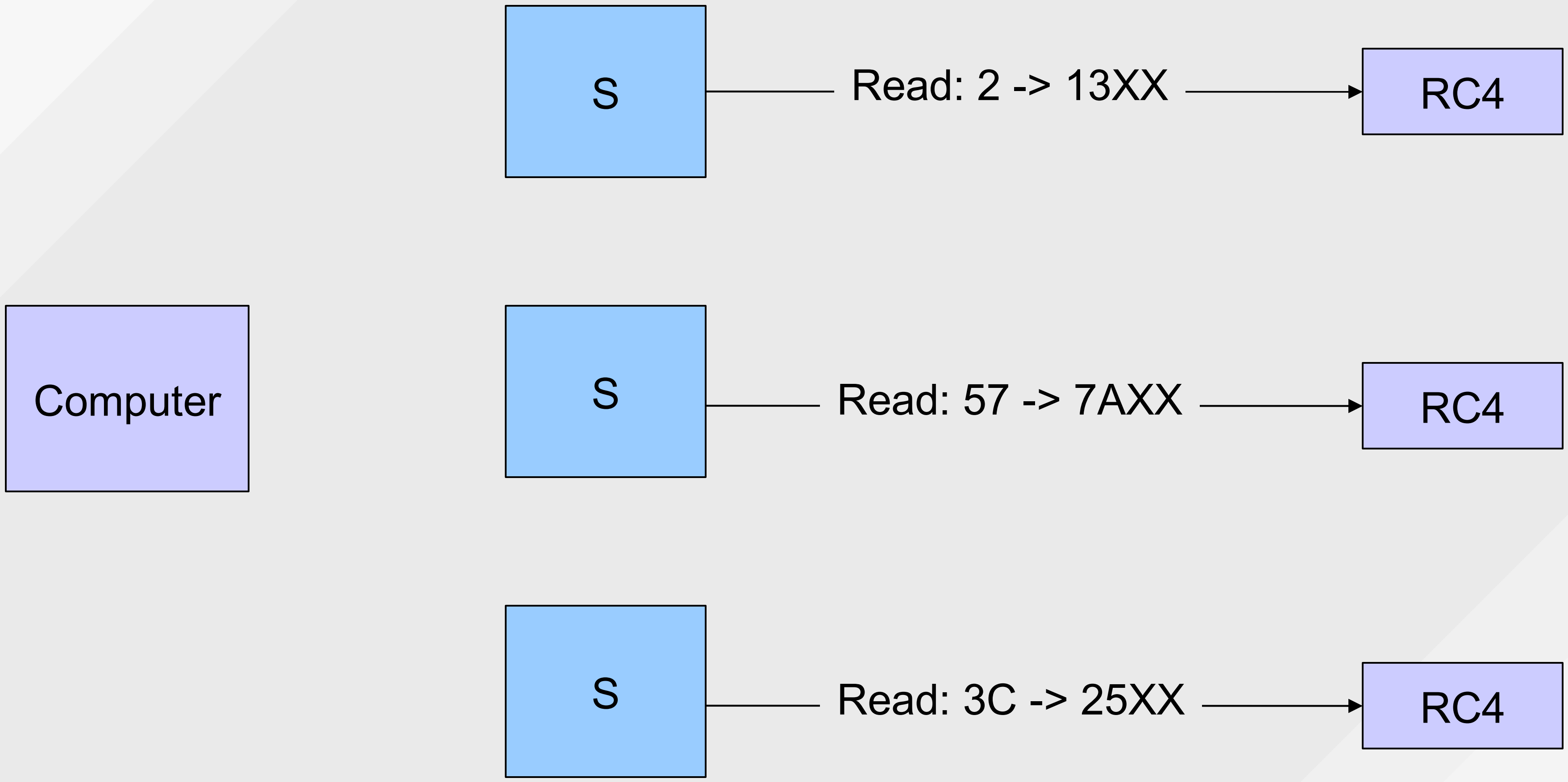
pico-wepcrack

00: 0073
01: 019B
02: 0296
03: 03c2
04: 0431
05: 05df
06: 0609
07: 078c
.....

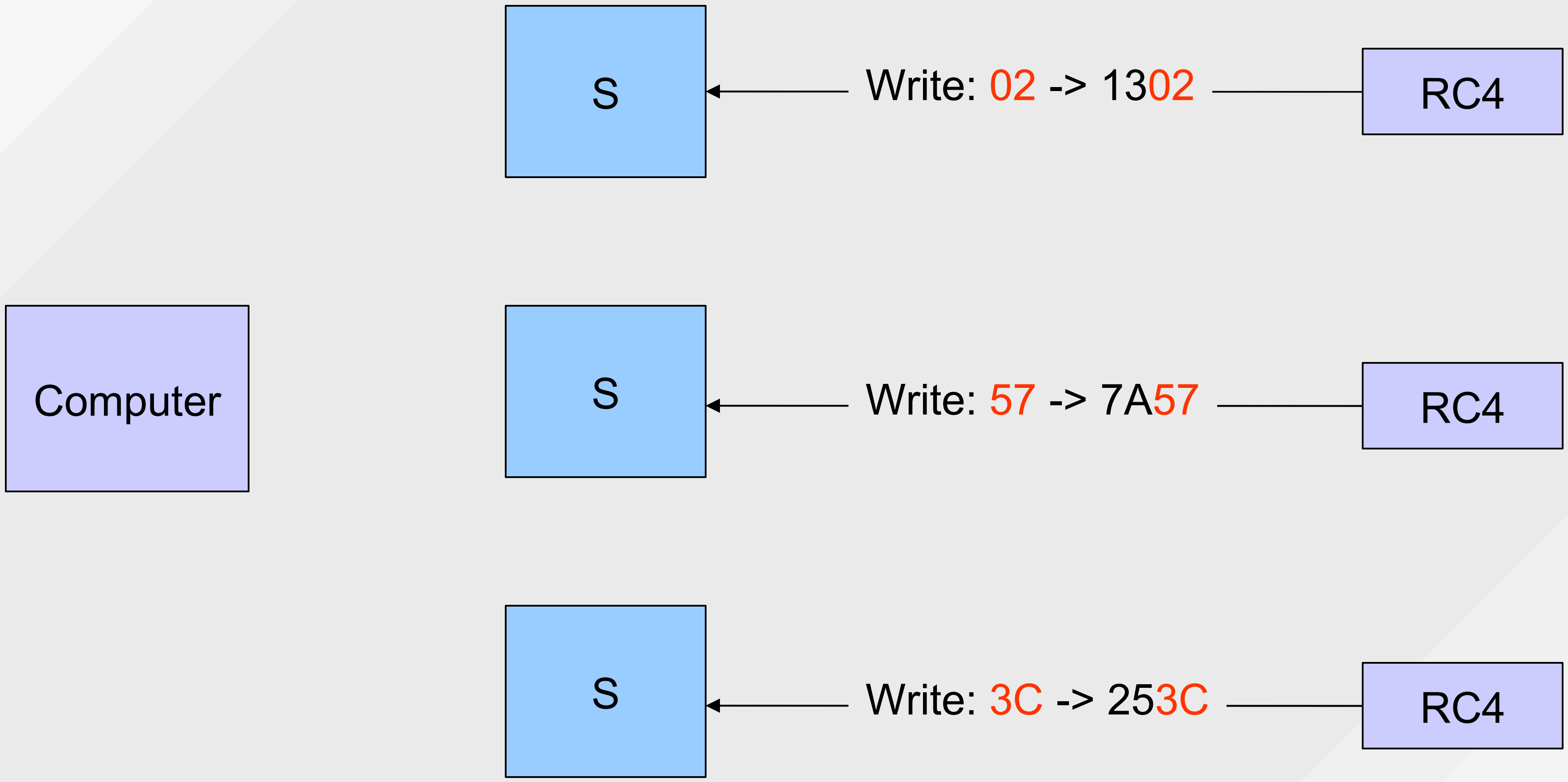
- RC4:

```
for(i = 0; i < 256; i++)                         // Init
   S[i] = i;                                     // S-Box must
   be reset

for(i = j = 0; i < 256; i++) {                   // KSA
   j += S[i] + K[i];
   Swap(S[i], S[j]);
}

for(i = 1, j = 0; ; i++) {                       // PRGA
   j += S[i];
   Swap(S[i], S[j]);
   PRGA[i - 1] = S[S[i] + S[j]];
}
```

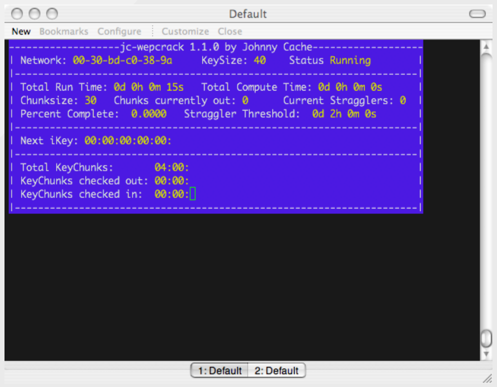
pico-wepcrack



pico-wepcrack



pico-wepcrack



```
------------------jc-wepcrack 1.1.0 by Johnny Cache------------------
| Network: 00-30-bd-c0-38-9a     KeySize: 40    Status Running      |
|-------------------------------------------------------------------|
| Total Run Time: 0d 0h 0m 15s   Total Compute Time: 0d 0h 0m 0s    |
| Chunksize: 30   Chunks currently out: 0      Current Stragglers: 0 |
| Percent Complete:  0.0000   Straggler Threshold:  0d 2h 0m 0s     |
|-------------------------------------------------------------------|
| Next iKey: 00:00:00:00:00:                                        |
|-------------------------------------------------------------------|
| Total KeyChunks:        04:00:                                    |
| KeyChunks checked out: 00:00:                                     |
| KeyChunks checked in:  00:00:                                     |
|-------------------------------------------------------------------|
```

+

+

Demo

OpenCiphers

Performance Comparison

PC		FPGA	
jc-wepcrack		pico-wepcrack	
1.25GHz G4	~150,000/sec	LX25	~9,000,000/sec
3.6GHz P4	~300,000/sec	15 Cluster	~135,000,000/sec
		FX60	~18,000,000/sec

Conclusion

- Get an FPGA and start cracking!
- Make use if your hardware to break crypto
- Add cool ascii matrix fx when you can :-)
- Choose bad passwords (please!)

Hardware Used

- Pico E-12
  - Compact Flash
  - 64 MB Flash
  - 128 MB SDRAM
  - Gigabit Ethernet
  - Optional 450MHz PowerPC 405

Hardware Used

- Pico E-12 Super Cluster
  - 15 - E-12’s
  - 2 - 2.8GHz Pentium 4’s
  - 2 - 120GB HDD
  - 2 - DVD-RW
  - 550 Watt Power Supply

OpenCiphers

Greetz

- Johny Cache (airbase/jc-wepcrack/jc-aircrack)
- Josh Wright (cowpatty)
- RenderMan (pmk hashtable monkey)
- Beetle (ghettopmk!)
- Audience (feel free to throw rotten fruit now!!)

Questions?

- I’ll give you a free set of hash tables!

- David Hulton
  - dhulton@openciphers.org
  - http://www.openciphers.org
  - http://www.picocomputing.com
  - http://www.802.11mercenary.net
  - http://www.churchofwifi.org