Algorithm 1 General Fingerprinting

1: **Input:** $Web\,browser\,instance$
2: **Output:** $Unique\,browser\,fingerPrint\,F$
3: $Initialise$ an empty $fingerPrint$ vector
4: Get array of functions as G
5: **for** i=1 to $G.length$ **do**
6: Add $G[i]$ result to the $fingerPrint$ vector
7: **end for**
8: Hash the fingerprint vector G to generate a unique fingerprint value F
9: **return** F

Algorithm 2 Software emulator for evaluating fingerpring detection method accuracy

Input:
N - the number of virtual devices for testing
$browser\_set[\,]\,(optional)$ - the list of browsers for testing
$user\_agent\_set[\,]\,(optional)$ - the list of user agents for testing

Output: $method\_accuracy\_n$

Define:

```
testing_report[{
    device_id,
    browser,
    user_agent,
    method_1_fingerprint,
    method_n_fingerprint
}]
```

for $i = 0$ to N **do**
  for $j = 0$ to $browser\_set.length$ **do**
    for $k = 0$ to $user\_agent\_set.length$ **do**
      Run: $browser[browser\_set[j]]$ with $user\_agent[user\_agent\_set[k]]$
      for $p = 0$ to n do **do**
        Run: pages with $finger\_print$ detection
        Write: testing report:
        $testing\_report \leftarrow i, browser\_set[j], user\_agent\_set[k], fingerprint\_n$
      end for
    end for
  end for
end for

Define:

```
error_rates[]
```

for $i = 0$ to N **do**
  Group results by $device\_id$:
  $grouped\_report = testing\_report.filter(val => val.device\_id === i)$
  Find the most frequent $fingerprint$ and its share from the total:
  $correct\_detections$
  Calculate device error:
  $Error\_n \leftarrow (100 - (testing\_report.length - correct\_detections))$
end for

Calculate method accuracy:
$method\_accuracy\_n \leftarrow (100 - \sum(error\_rates[n])/\,(error\_rates.length))$

return $method\_accuracy\_n$