Lightweight Cryptography

Dr Pekka Jäppinen

Lappeenranta University of Technology

Outline

- Background
- What is lightweight
- Metrics
  - Chip area
  - Performance
- Implementation tradeoffs
- Current situation
- Conclusions

Background

- Emergence of ubiquitous Computing
  - Computing and communication capabilities are implemented in ever smaller and less powerful devices.
- Privacy and security concerns are affecting on acceptance of new technologies
  - Liability issues in case of information leak
  - European Comission joint research center: RFID Technologies: Emerging Issues, Challenges and Policy Options :http://www.jrc.es/publications/pub.cfm?id=1476

- Traditional security primitives requirements are too high for simple systems
  - Sensor networks, RFID tags.
  - Martin Feldhofer, Christian Rechberger: A case against currently used Hash functions in RFID protocols
- Comparing the weight is hard
  - Metrics for weight of algorithms are provided in variety of degree
  - Metrics for protocols are practically non existent

What is Lightweight Cryptography

- Lightweight solutions is designed by keeping in mind the restriction of small devices...
  - Computational power, Memory, Storage space, Available energy
- ...While maintaining adequate performance
- Lightweight does not imply less secure
  - Goal is to have lightweight solutions as secure as heavyweight solutions
  - Compromises may be needed.

- There is no official definition when solution can be called as lightweight
  - Everyone has their own perspective
  - Depends on context the term is used
    - What might be lightweight for software is not necessarily lightweight for hardware and vice versa
    - What is lightweight for PC is not necessarily lightweight for RFID
- This presentation focuses on the lightweight as suitable for smallest of devices e.g. RFID and sensors

Metrics

- In order to compare solutions we need metrics
- Goals for lightweigt solution
  - Cheap to build
  - Fast
  - Requires little power
- Martin Feldhofer, Johannes Wolkerstorfer: Strong Crypto for RFID Tags - A Comparison of Low-Power Hardware Implementations
  - Chip area, Clock cycles, power consumption

Metrics: Chip area

- Required chip area can be estimated in terms of required logical (NAND) gates
  - The more gates are needed the more expensive the solution will be.
  - More gates requires more power
- Simpler the tag higher the proportional cost from security
  - EPC tag has ~10000 Gates from which ~2000 could be reserved for security (Juels and Weis) (25% extra)
  - 1000 Gates cost approximately 1 us cent (0.3

Where all those gates go

- Memory gates
  - Gates needed for storing data like pseudonyms, challenges, random numbers, history data, middle results like chaining vectors etc.
- Processing gates
  - Gates needed for algorithms, random number generation, mathematical functions etc.
- Communication gates
  - mainly buffers

Some Gate counts

- Storage: 8GE/bit (temporary), 3GE/bit (longterm, conservative approximation)
- Hashes
  - Sha1: 8120 GE, SHA256 10868 GE
- Symmetric Crypto
  - AES-128: 3400 GE, DESL: 1848
- Asymmetric crypto
  - ECC-192: 23600, WIPR: 5705

Performance

- Longer activity time increases required power
- RFID tag has to respond to the reader within certain time
  - Security may not slow the system down more than this
- Slowness in response easily accumulates if there are hundreds or thousands of small devices to communicate with.
- The speed of implementation is dependent on the clock speed of the platform it is run.
  - Comparing the performance of algorithms can be

Some Clock cycles

- Storage: 1CC
- Hash:
  - SHA-1: 1274 CC, SHA-256: 1128 CC
- Symmetric crypto
  - AES-128: 1632 CC, DESL 144 CC
- Asymmetric crypto
  - WIPR: 66048

Performance and protocols

- For simple request-reply solutions the differences in communication time is insignificant
  - The amount of data transferred is not very big
  - Actual time depends on used communication system.
- Some solutions transfer computational complexity to communication complexity.
  - e.g Probabilistic authentication use several challenge-response pairs for authentication. → Low GE and CC
    - latency slows down the solution -> multiply the CC used for generating response for approximation

- performance vs security
  - In probabilistic authentication the more rounds you have the longer the authentication takes but more sure you can be on the authenticity
    - 1 challenge 50%, 2 challenges 75%, 3 challenges 87.5% ...

Parallel vs serial implementation

- Parallel is faster than serial
  - More operations / clock cycle
- Parallel solution often requires more gates
  - Gates are not reused
  - Serial solution need additional shift registers for control.
- Parallel requires more energy / clock cycle, serial requires more total energy
  - RFID tags that get power from reader has limit on energy/ clock cycle

Asymmetric weight

- Communicating devices may have different computational capabilities
  - Mobile phone – desktop computer
  - RFID tag – RFID backend server
- Put the more powerful device do all the hard calculations
  - In RSA you can select encryption power so that encryption is simple, while decryption requires more power
  - Use of random small seed on weak device for additional entropy

Current situation

- Lightweight solutions researched mainly for RFID
- Learn more on solutions:
  - Gildas Avoine RFID security and privacy lounge at http://www.avoine.net/rfid/ contains list of research papers around RFID security and privacy, newest first.
  - Ari Juels: RFID Security and Privacy: A research Survey
  - Selwyn Piramuthu: Protocols for RFID tag/reader authentication

Conclusions

- Lightweight solutions are needed when we surround ourselves with more computing devices
- Solution is not lightweight just if the developer says so.
  - So far there are no lightweight hash functions (that I know of).
  - It is possible (and important) to estimate the basic weight metrics of the protocol without doctorate degree in digital electronics.
- Used platform and type of use affects what would be optimal solution

References

- Martin Feldhofer and Christian Rechberger. A Case Against Currently Used Hash Functions in RFID Protocols. In Proceedings of First International Workshop on Information Security (IS'06), volume 4277 of LNCS, pages 372–381, 2006.
- Denis Trcek and Damjan Kovac. Formal apparatus for measurement of lightweight protocols. Computer Standards and Interfaces, 31(2):305–308, February 2009
- Martin Feldhofer and Johannes Wolkerstorfer. Strong Crypto for RFID Tags – A Comparison of