XIX IMEKO World Congress
Fundamental and Applied Metrology
September 6–11, 2009, Lisbon, Portugal

ALGORITHMS AND CIRCUITS FOR LOW POWER SECURED SENSOR NETWORKS WITH ASYMETRIC COMPUTATIONAL RESOURCES

Tomasz Adamski [1], *Wiesław Winiecki* [2], *Jakub Olszyna* [3]

[1] Warsaw University of Technology, Institute of Electronic Systems, Warsaw, Poland,T.Adamski@ise.pw.edu.pl
[2] Warsaw University of Technology, Institute of Radioelectronics, Warsaw, Poland, W.Winiecki@ire.pw.edu.pl
[3] Warsaw University of Technology, Institute of Radioelectronics, Warsaw, Poland, J.Olszyna@ire.pw.edu.pl

Abstract – The paper deals with applications of cryptographic methods in design of secured versions of Distributed Measurement Systems (DMS). In the paper wide range of cryptographic algorithms is assessed and some of them are chosen and proposed as well suited to DMS specific requirements. Some specialized low power, high speed, hardware solutions for cryptographic algorithms are also suggested in the paper. Strong asymmetry of computing power and memory capacity is taken into account in both software and hardware solutions.

Keywords: sensor networks, information security, cryptographic systems, low power circuits

1. INTRODUCTION

In many kinds of Distributed Measurement Systems (DMS) information security of the system is the crucial design problem. Additionally in many instances of DMS, like sensor networks (SN), wireless sensor networks (WSN) and mobile DMS applications we deal with tiny autonomous nodes with very limited computational resources and extremely limited energy source. These constraints influence security solutions (algorithms, protocols, circuits) which can be used in such networks.

Specific properties of DMS considered in the paper are the following:

1.asymmetry of computational resources (for example computational power and memory capacity are very limited in tiny autonomous sensor nodes)

2. small bandwidth of data transmission channels of the network

3. energy constraints (typically we can obtain about 10-50 μW from the scavenger which produces supply voltage relying solely on ambient vibrations)

4. use of unsecured channels (Internet or public telecommunication channels)

The four fundamental security services are the following: privacy, entity authentication message authentication and data integrity.

Our aim is to propose security solutions (algorithms, protocols, circuits) for these services which are well suited to above mentioned DMS specific requirements.

The security aspects of low power DMS networks with asymetric computational resources are a very active topic of research with far reaching applications (from environment monitoring, collecting microclimate data to several military applications like target tracking and detecting bio-weapons [7], [8], [11], [12]). Special attention in recent works is also payed to self-powered sensor networks. Emerging ultra low power DMS networks and massive measurements need special security methods, algorithms and circuits.

2. CIPHERING ALGORITHMS

Most publications seem to preclude that Public Key Cryptography (PKC) is not feasible on severly resource constrained sensor nodes. In this paper we show that PKC simplifies the implementation of many typical security services and additonally reduces transmission power due to less protocol overhead.

2.1. Rabin algorithm

Rabin algorithm (or Rabin's cipher) is a classical public key cipher with provable security. Rabin's cipher security is based on the factorization problem of large numbers and is therefore similar to the security of RSA with the same size modulus. Rabin's cipher has asymetric computational cost. The encryption operation is very simple and faster than decryption. Its asymetry makes Rabin's cipher an interesting choice for sensor networks in which nodes and base stations (servers) have different computational capabilities

Key generation for Rabin public key encryption is the following. Every entity generates two large random (and distinct) primes *p*, *q* and computes $n = p \cdot q$. An ordered pair (p, q) is a private key of the entity and $n = p \cdot q$ is a public key. It is useful for simplicity of decryption algorithm to choose such primes p an q that $p \equiv 3(\bmod 4)$ and $q \equiv 3(\bmod 4)$.

Assume $m \in Z_n$ is a plain text message where Z_n is a ring of integers modulo *n*. The cryptogram is given as

$c = m^2 (\mathrm{mod}\, n)$ then to encrypt the message we need only one multiplication modulo *n*.

Decryption algorithm is the following. Assume we have a cryptogram $c = m^2 (\mathrm{mod}\, n)$ and $p \equiv 3 (\mathrm{mod}\, 4)$**,** $q \equiv 3 (\mathrm{mod}\, 4)$**.** To calculate *m* from *c* we have to compute 4 possible square roots m_1, m_2, m_3, m_4 and choose from them an appropriate plain text message *m*. Square roots from the cryptogram *c* can be computed in the following way:

1. We find integers $a, b \in Z$, that $a \cdot p + b \cdot q = 1$ (using for example the extended Euclid algorithm) and next we compute 4 numbers

$$r = c^{(p+1)/4} (\mathrm{mod}\, p) , \qquad s = c^{(q+1)/4} (\mathrm{mod}\, q) ,$$
$$x = (aps + bqr)(\mathrm{mod}\, n) , \quad y = (aps - bqr)(\mathrm{mod}\, n) ,$$

2. Four square roots are the following

$$x, -x (\mathrm{mod}\, n) , \quad y, -y (\mathrm{mod}\, n) .$$

The above decryption algorithm is complicated but it is implemented in servers (base stations) without severe constraints on resources.

The attack on the Rabin algorithm consists in recovering plaintext *m* from the corresponding cryptogram $c = m^2 (\mathrm{mod}\, n)$. This is precisely the problem of the square root modulo *n* which is computationally equivalent (it can be proved) to the problem of factoring *n* . Finally, security of the Rabin public-key encryption algorithm is based on hardness of factoring *n*.

2.2. RSA algorithm with small encryption exponents

The RSA cipher is widely used in computer networks but has relatively large computational complexity and can be used in DMS with resource constrained nodes only for rather small public keys for example 3, 5, 7.

The RSA encryption scheme is the following. Each entity creates an RSA public key *e* and a corresponding private key *d*. A public key is a number $e \in Z_{\varphi(n)}$, where $Z_{\varphi(n)}$ is a ring of integers modulo $\varphi(n)$ and $\varphi(n)$ is the Euler function value for the argument $n = p \cdot q$, where *p, q* are diffrent primes . We assume that $GCD(e, \varphi(n)) = 1$ i.e. $GCD(e, (p-1)(q-1)) = 1$. The private key is a number $d \in Z_{\varphi(n)}$, which is an inverse of $e \in Z_{\varphi(n)}$ in the ring $Z_{\varphi(n)}$. The plain text messages are elements of the ring Z_n .

The cryptogram is given as $c = m^e (\mathrm{mod}\, n)$. The plain text message can be obtained from the cryptogram *c* and the private key *d* by the formula $m = c^d\, (\mathrm{mod} n)$.

The RSA cipher with small exponents (small ciphering keys *e*) is an interesting choice because we have only one solution $m \in Z_n$ of the equation $c = m^e (\mathrm{mod}\, n)$ (if $GCD(e, (p-1)(q-1)) = 1$) and decrypted message is unique.

There are some known attacks on RSA with small ciphering public keys then such systems have to be carefully designed. The attack on the RSA cipher with a small exponent *e* is for example possible if the same plain text message *m* is ciphered *e* times with the same public key *e* modulo $n_i \geq 2, n_i \in N$ with $GCD(n_i, n_j) = 1$ for $i \neq j$.

Rabin and RSA ciphers are block ciphers then using universal schemes (for example Rabin, Davies schemes see [1],[2]) we can in natural way design hash functions.

3. ENTITY AUTHENTICATION ALHORITMS

Entity authentication (or entity identification) is one of four main security objectives. It seems that the best solution for discussed DMS are so called zero-knowledge identification algorithms (or protocols). In the sequel the idea of zero-knowledge proof is shortly explained.

3.1. Zero knowledge proof

The simplest example which explains very well the idea of the zero-knowledge proof is so called “zero-knowledge cave” (fig.1). We have two entities Prover and Verifier. Prover wants to prove that he has a key to the normally closed door in the zero-knowledge cave but he doesn’t want to show the key. Prover and Verifier play the following game.

1.Prover goes to the cave door and Verifier stays outside.

2. Verifier goes to the cave point denoted with arrows and ask Prover to come out from the left side (or right side with equal to ½ probability).

If Prover goes from the left side (or respectively from the right side) it suggests that Prover has the key which opens the internal cave door. If the verifier order is not fulfilled correctly the protocol is stopped because Prover cheats. He has no key to the door. The above protocol is repeated *t* times. Probability of successful cheating by a Prover without the key is equal to $(1/2)^t$.

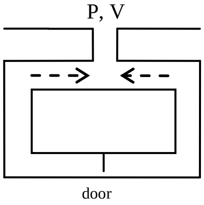

Fig.1 Zero-knowledge cave, *P* is a Prover, *V* is a Verifier

3.2. Fiat-Shamir protocol

The Fiat–Shamir protocol is a zero-knowledge proof used to entity authentication. The protocol is well suited to considered DMS networks. Basic version of the Fiat-Shamir identification (entity authentication) algorithm is the following.

A trusted center *T* selects and publishes an RSA like modulus $n = p \cdot q$ (where *p*, *q* are diffrent primes) and keeps prime *p* and *q* secret. Each prover *P* selects a secret number $s \in Z_n$ coprime to *n*, computes $v = s^2 (\mathrm{mod} n)$ and registers *v* as its public key.

Assume *P* (a prover) proves knowledge of *s* to *V* (a verifier). *P* does it in *t* executions of a 3-pass protocol.

1.*P* chooses a random number $r \in Z_n$ and sends to *V* a number $x = r^2 (\mathrm{mod} n)$ (*r* is so called “witness”).

2. *V* sends to *P* a random bit $e \in \{0,1\}$ (*e* is so called “challenge”).

3. *P* sends to *V* a number $y = r \cdot s^e (\mathrm{mod} n)$ (*y* is so called “response”)

The verifier *V* rejects the proof if $y^2 \neq x \cdot v^e (\mathrm{mod} n)$. Probability of successful cheating of the prover *P* (if he doesn’t know a secret number *s*) is even to $(1/2)^t$ where *t* is a number of protocol executions.

For hardware implementation of the above algorithm we need only a multiplier and a random bit generator.

Zero-knowledge protocols can be also used in message authentication alhoritms (digital signatures).

4. DOCUMENT AUTHENTICATION ALHORITMS

4.1. Digital signatures based on Fiat-Shamir protocol

Assume we have Fiat-Shamir protocol which generates sequences: witness $\rightarrow$challenge $\rightarrow$response. If $x = (x_1, x_2, ..., x_t)$ is a sequence of random numbers which are squares of random $r_i \in Z_n$ (r_i is a witness) and $e = (e_1, e_2, ..., e_t) = h(m)$, where *h* is a given hash function, $m \in Z_n$ is a signed document and $e = (e_1, e_2, ..., e_t) \in \{0,1\}^n$ is a sequence of challenges. The signer knows a secret $s \in Z_n$, he publishes $v = s^2 (\mathrm{mod} n)$ and computes the correct response $y_i \in Z_n$ for every $r_i \in Z_n$ and $e_i \in Z_n$. The signature of the document $m \in Z_n$ is an ordered pair:

$$(x, y) = ((x_1, x_2, ..., x_t), (y_1, y_2, ..., y_t)).$$

Someone who knows a public key $v = s^2 (\mathrm{mod} n)$ of the signer can verify the signature (x, y) veryfying every coordinate exactly like in the Fiat-Shamir protocol.

5. SPECIALIZED CRYPTOGRAPHIC CIRCUITS

5.1. Serial bit multiplication circuit

A proposed multiplication circuit is shown in the fig. 2. The circuit is a serial data word processing multiplier which minimalizes number of gates and multiplies two *k* bit numbers in k^2 clock cycles. All registers used in the system are shift registers. Proposed multiplication circuit was simulated in VHDL and tested. Obtained results allow to asses power consumption and prove that the multiplier works correctly.

5.2. Circuit for reduction modulo p based on Barrett’s algorithm

Assume $p \in N$, $p \geq 2$, $b \in N, b \geq 3$ and $k = \lfloor \log_b p + 1 \rfloor$. The Barrett’s algorithm (Barrett reduction) finds the value $z (\mathrm{mod}\, p)$, for a given $z \in \{0,1,\ldots,b^{2k} - 1\}$ and a modulus *p*. The Barrett algorithm does not exploit any special form of the modulus *p*.

There is a number $q \in N \cup \{0\}$ that we have $z = q \cdot p + r$, where $r \in N \cup \{0\}$ and $0 \leq r < p$ is a remainder. It is easy to verify that $q = \lfloor z / p \rfloor$ then $z = \lfloor z / p \rfloor p + r$. Define an algorithm constant $\mu = \lfloor b^{2k} / p \rfloor$ which can be written with a $(k+1)$ bit word. The Barrett’s algorithm is the following.

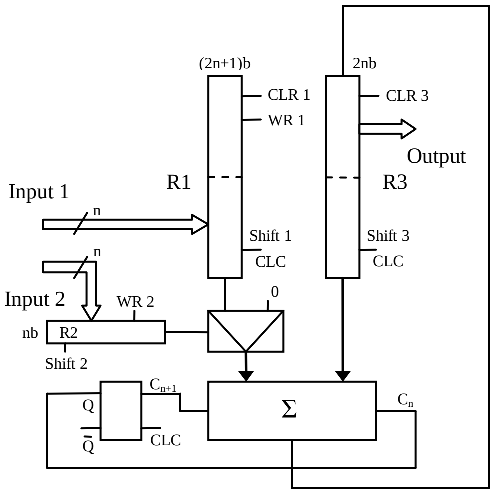

Fig. 2. Serial multiplication circuit, R1, R2, R3 are shift registers.

```
Barrett's algorithm

Input Data: $p \in N, p \geq 2$, $b \in N, b \geq 3$,
$k = \lfloor \log_b p + 1 \rfloor$, $z \in <0, b^{2k} - 1>$ and $\mu = \lfloor b^{2k} / p \rfloor$

Output Data : $z(\mathrm{mod}\, p)$

1. $\hat{q} := \lfloor \lfloor z / b^{k-1} \rfloor \cdot \mu / b^{k+1} \rfloor$
2. $r := z(\mathrm{mod}\, b^{k+1}) - \hat{q} \cdot p(\mathrm{mod}\, b^{k+1})$
3. if $r < 0$ then $r := r + b^{k+1}$
4. while $r \geq p$ do $r := r - p$
   return($r$)
```

Fig. 3. The Barrett's algorithm.

The Barrett's algorithm (fig.3) computes the value $z(\mathrm{mod}\, p)$ with 2 multiplications by a constant and 3 additions/subtractions. The divisions required in the algorithm are simple shifts of the base b representations. A natural choice for the base is a power of 2. Integers p and z are large in cryptographic applications (with 100-300 decimal digits). Steps 3 and 4 in the algorithm are correction steps. Only one correction is possible in the step # 3 and two corrections in the step # 4.

Fact (correctness of the Barrett algorithm)
If $p \in N, p \geq 2$, $b \in N, b \geq 3$, $k = \lfloor \log_b p + 1 \rfloor$, $\mu = \lfloor b^{2k} / p \rfloor$, $z \in <0, b^{2k} - 1>$ then the Barrett algorithm is correct i.e. finds $z(\mathrm{mod}\, p)$.

Proof. Proof of this fact starts from a simple equality

$$\frac{z}{p} = \frac{z}{b^{k-1}} \cdot \frac{b^{2k}}{p} \cdot \frac{1}{b^{k+1}}$$

Next we prove that $\hat{q} := \lfloor \lfloor z / b^{k-1} \rfloor \cdot \mu / b^{k+1} \rfloor$ is a good approximation of the quotient $q = \lfloor z / p \rfloor$. The full proof that the Barrett's algorithm is correct is given in [1] and [6]. ■

In the paper the Barrett algorithm is mapped to a systolic array shown in the fig. 4. Proposed systolic structure computes the value $z(\mathrm{mod}\, p)$ in $7k^2$ clock cycles. The rectangular clock signal denoted as CLK' is k times slower than CLK. Every block of the circuit corresponds to one step of the Barrett algorithm. Decision blocks # 1 (step 3) and decision block number 2 (step 4) are shown in the fig. 5 and 6 respectively. The circuit can be easily modified to compute values $a^2(\mathrm{mod}\, p)$ and $a \cdot b(\mathrm{mod}\, p)$. The proposed reduction modulo p circuit was simulated in VHDL and tested. Obtained results will be discussed in the full paper. In VHDL simulation we assume the base $b = 4$. Multiplication with this base is equivalent to common binary multiplication then we can use a typical binary multiplier at two first levels of the systolic array.

Multiplication modulo p can be also implemented using well known Montgomery reduction and multiplication algorithms but proposed circuits based on the Barrett algorithm seems to be more flexible and universal. Both algorithms show similar execution times and similar area on chip.

6. CONCLUSIONS

1. We have proposed in the paper algorithms and specialized cryptographic circuits which can be used in low power, secured Distributed Measurement Systems with asymmetric resources like for example sensor networks.

Chosen cryptographic algorithms for the host with limited resources are secure but as simple as possible from computational point of view. Distributed Measurement Systems based on public key cryptography minimalize key distribution and key management problems.

2. We have showed, that it is possible to implement all crucial kinds of cryptographic algorithms (for privacy, entity authentication, message authentication, data integrity and random bit generation) with two simple circuits (building blocks) with bit serial word processing: multiplier and systolic circuit for the Barrett algorithm .

3. In our assessment we have obtained a good balance between high-speed and low-power capabilities of proposed solutions.

REFERENCES

[1] A. Menezes, P. Oorschot, S. Vanstone, *Handbook of Applied Cryptography*, CRC Press Inc., 2001

[2] J.A.Buchmann, *Introduction to Cryptography*, Springer Verlag, New York, 2004.

[3] D. Richter, "IT Integration in Metrology", *XVIII IMEKO World Congress*, pp. 1-5, Rio de Janeiro, Brazil, Sept. 2006.

[4] D. R. Stinson; *Cryptography- Theory and Practice*, CRC Press Inc., 2002.

[5] L.Buttyan, V. Gligor, D. Westhoff (Eds), *Security and Privacy in Ad-Hoc and Sensor Networks*, LNCS, Springer Verlag, Heidelberg ,2006.

[6] D. Hankerson, A. Menezes, S.Vanstone, *Guide to Elliptic Curve Cryptography*; Springer Verlag, New York, 2004.

[7] G. Gaubatz, J. Kaps, B. Sunar, "Public Key Cryptography in Sensor Networks – Revisited", http://www.crypto.wpi.edu.

[8] K. Yuksel, J. Kaps, B. Sunar, "Universal Hash Functions

For Emerging Ultra-Low-Power Networks”, (YukselKapsCnds04.pdf) http://www.crypto.wpi.edu.

[9] M.Heger, “Cryptographers Take On Quantum Computers”, *IEEE Spectrum*, p 10, January 2009.

[10] J. Pieprzyk, T. Hardjono, J. Seberry, *Theory of Computer Systems Security*, Springer Verlag, Heidelberg, 2007.

[11] G. Gaubatz, J. Kaps, E. Ozturk, B. Sunar, “State of the Art in Ultra-Low Power Public Key Cryptography for Wireless Sensor Networks”, http://www.crypto.wpi.edu.

[12] T.Adamski, W. Winiecki, “Entity Identification Algorithms for Distributed Measurement and Control Systems with Asymmetry of Computational Power”, *Przegląd Elektrotechniczny*, pp. 216-220 (in Polish), Vol. 2008, No 5.

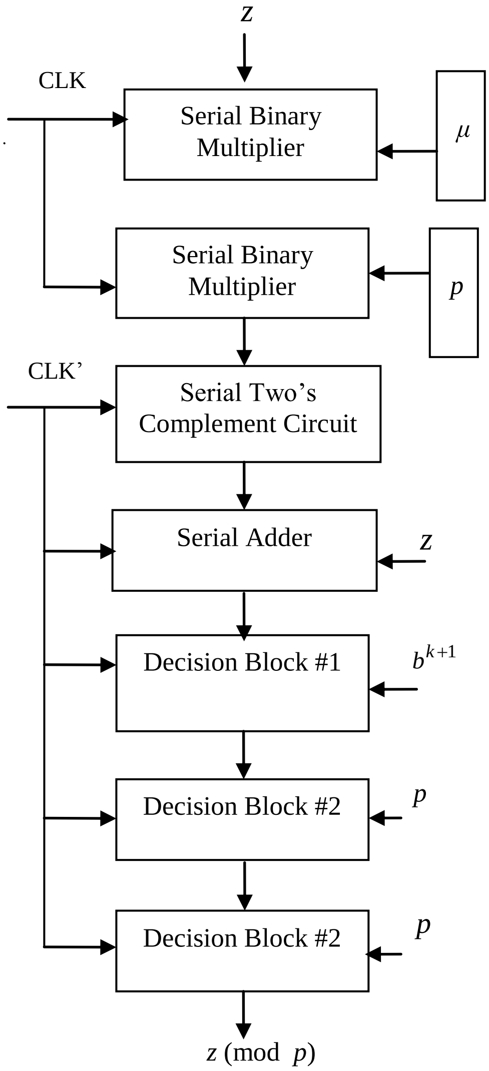

Fig. 4 Implementation of the Barrett’s algorithm. The systolic circuit computes $z(\bmod p)$.

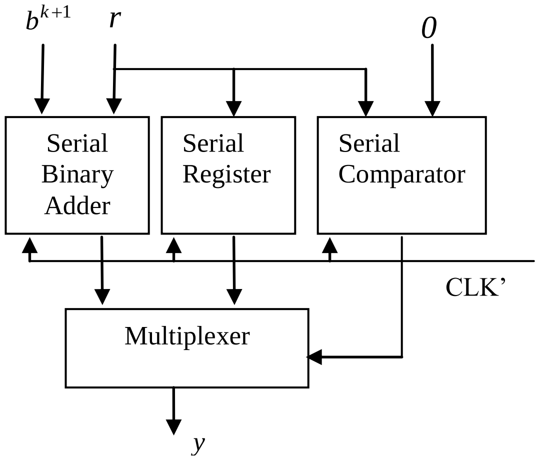

Fig. 5 Decision block # 1 conditionally adds b^{k+1} to the input value r if $r < 0$.

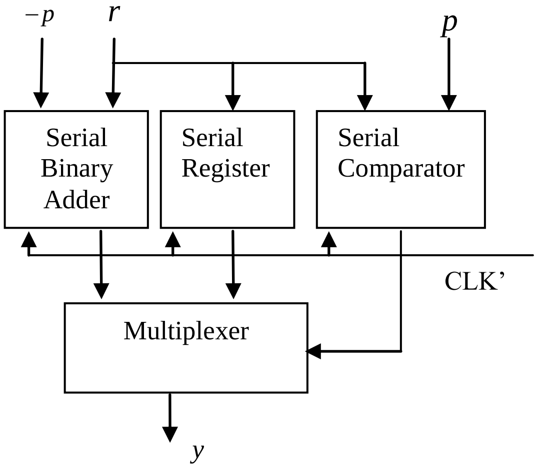

Fig. 6 Decision block # 2 subtracts p from the input value i.e. adds $-p$ (two’s complement of p) to the input value r if $r > p$.