APPLICABILITY OF MULTIPARTY COMPUTATION SCHEMES FOR WIRELESS SENSOR NETWORKS
Position Paper

Manuel Koschuch, Matthias Hudler, Michael Krüger
Competence Centre for IT-Security, FH Campus Wien, University of Applied Science, Favoritenstrasse 226, Vienna, Austria

Peter Lory
Institut für Wirtschaftsinformatik, Universität Regensburg, Universitätsstrasse 31, Regensburg, Germany

Jürgen Wenzl
TMMO GmbH, Vilsgasse 25, Kallmünz, Germany

Keywords: Sensor networks, Threshold cryptography, Efficient implementation, Multiparty computations.

Abstract: Wireless Sensor Networks pose special requirements to the deployed security algorithms, due to their unique properties: a single sensor node has great restrictions in terms of computing power, available memory and available energy. It is nevertheless desirable for the messages exchanged over the air interface to be secure against eavesdropping and forging. Since a single sensor node can be captured and removed very easily and almost undetectable, cryptographic schemes that do not rely on a single master secret present in every node, like those based on, for example, multiparty computations, seem to be a promising alternative in this setting. We are currently investigating the applicability of a modified implementation of the Gennaro-Rabin-Rabin multiparty multiplication protocol for sensor networks, with a special focus on the number of messages that have to be exchanged and the additional load put on every node by this protocol. This paper gives a short overview of our work and lists some preliminary results.

1 INTRODUCTION

Wireless Sensor Networks can be used for a wide variety of applications: from environmental monitoring to energy plant surveillance, the huge number of inexpensive sensors, covering a wide area and communicating measurements in a hop-to-hop fashion allows many new approaches not suitable for common network infrastructures. The main problem when designing and deploying such networks lies in the security requirements: messages are transmitted over the air interface and are thus susceptible to eavesdropping, modification or forging. To prevent such attacks, suitable cryptographic protocols and algorithms have to be employed. The majority of currently deployed sensor networks use symmetric cryptography, due to the resource constraints of the individual nodes, to secure the communication between the network elements, although the suitability of asymmetric cryptography utilizing elliptic curves for this environment is a big current research topic.

When using symmetric cryptography, the problem of key distribution arises: what mechanisms are employed to ensure that two adjacent nodes share a common key? Giving every node in the entire network the same key has obvious security implications, since capture and analysis of a single node (which is usually easy to do and stays almost completely undetected, given a large enough network) compromises the key used in the entire network.

Another approach is the use of heuristic algorithms, where every node is preloaded with a subset of keys taken from one large keypool, hoping that when trying to communicate with an adjacent node both nodes find one common key in their respective sets. Using this method, there is no single key for the entire network, which an attacker could obtain from capturing a single node, although he can still compromise a certain subset of the network (see also (Merwe et al., 2007) for an overview of different key manage-

Koschuch M., Hudler M., Krüger M., Lory P. and Wenzl J. (2010).
APPLICABILITY OF MULTIPARTY COMPUTATION SCHEMES FORWIRELESS SENSOR NETWORKS - Position Paper.
In *Proceedings of the International Conference on Data Communication Networking and Optical Communication Systems*, pages 125-128
DOI: 10.5220/0003032901250128

ment techniques for wireless sensor networks).

In this setting, the use of some sort of threshold cryptography seems promising: instead of relying on the integrity of every single node, a certain number of uncompromised nodes is required to produce a valid result, and no single node knows the entire secret required to produce this result. The amount of nodes that have to cooperate in order to create the secret is equivalent to the number of nodes an attacker has to successfully compromise before gaining access to the communication in the network.

Our work now tries to quantify the applicability of threshold cryptography for wireless sensor networks in general, with a first focus on multiparty computations utilizing the Gennaro, Rabin and Rabin (Gennaro et al., 1998) protocol, with some numeric optimizations. The remainder of this paper is structured as follows: Section 2 gives a general introduction to Multiparty computations and the Gennaro, Rabin and Rabin protocol. Section 3 then sums up our current experimental results, while finally Section 4 details our next steps.

2 MULTIPARTY COMPUTATIONS

Protocols for multiparty multiplication of two polynomially shared values over $\mathbb{Z}_q$ with a public prime number q are important cryptographic primitives in various application fields.

Polynomial sharing refers to the threshold scheme originally proposed by Shamir (Shamir, 1979), which assumes that n players share a secret α in a way that each player P_i ($1 \leq i \leq n$) owns the function value $f_\alpha(i)$ of a polynomial f_α with degree at most t and $\alpha = f_\alpha(0)$. Then any subset of $t+1$ participants can retrieve the secret α (for example by Lagrange's interpolation formula) but no subset of, at most, t participants can do so.

At the beginning of the multiplication protocol each player P_i holds as input the function values $f_\alpha(i)$ and $f_\beta(i)$ of two polynomials f_α and f_β with maximum degree t and $\alpha = f_\alpha(0), \beta = f_\beta(0)$. At the end of the protocol each player owns the function value $H(i)$ of a polynomial H with maximum degree t as his share of the product $\alpha\beta = H(0)$.

Lory (Lory, 2007) and (Lory, 2009) has presented protocols for this task. They accelerate the technique of Gennaro, Rabin and Rabin (Gennaro et al., 1998), which was known for its efficiency among its contemporary competitors (see e. g. Cramer and Damgård (Cramer and Damgård, 2005)). All these protocols consist of two steps. In a first step, each player P_i with $1 \leq i \leq 2t+1$ computes $f_\alpha(i) f_\beta(i)$ and shares this value with the other participants using a polynomial $h_i(x)$ of maximum degree t. He sends player P_j with $1 \leq j \leq n$ the value $h_i(j)$. Here, it is assumed that the n parties with $n \geq 2t+1$ are connected by secure point-to-point channels. When used in the environment of sensor networks, this task could be done when producing the actual sensor nodes, before deployment into the field.

In a second step, each of these players computes his share $H(j)$ of $\alpha\beta$ by combining the values $h_i(j)$ for $i = 1, 2, \ldots 2t+1$. The approach is (unconditionally) secure against an adversary, who can corrupt at most t of the players under the so-called "honest-but-curious" model. This means that the adversary is passive and can read the memories of the corrupted players but not modify their behavior. For details the reader is referred to the original papers.

The first step of the multiplication protocol of Gennaro, Rabin and Rabin (Gennaro et al., 1998) requires $O(n^2 k \log n)$ bit-operations per player, where k is the bit size of the prime q and n is the number of players.

In the corresponding modified step of (Lory, 2007) this complexity is reduced to $O(n^2 k)$. The second step of the protocol in (Gennaro et al., 1998) requires $O(nk^2)$ bit-operations per player. The corresponding step in (Lory, 2009) has a complexity of $O(n^2 k)$. Of course, the latter is an improvement only, if the number of players is considerably smaller than k. This is true in many cases, because $k \geq 1024$ in many practical situations. All the protocols need one round of communication (in the first step).

The above complexities are valid under the assumption that all multiplications are performed in the classical manner, i. e. a multiplication of an l_1-bit-integer and an l_2-bit-integer requires $O(l_1 l_2)$ bit-operations. This is realistic, if the bit-lengths are not too large. For very large numbers, other methods like the algorithm of Karatsuba, the Toom–Cook algorithm or discrete Fourier transformation based algorithms are faster (see Knuth (Knuth, 1998)). Careful numerical experiments by Wenzl (Wenzl, 2010), whose implementation was the base for our research, demonstrate, that also in these cases considerable reductions in computing time can be achieved by the methods of (Lory, 2007) and (Lory, 2009).

3 PRELIMINARY RESULTS

In our first approach we were interested in two things: how does the improved protocol scale in comparison to the unmodified Gennaro, Rabin, and Rabin version

when altering the number of players and the number of bits in the underlying field, and how many cycles are used on actual hardware. The following results were obtained using the software implementation from (Wenzl, 2010). For the required long integer arithmetic, the GNU multiple precision arithmetic library [1] in version 5.0.1 was utilized. We ran the implementation on an AMD Athlon64 X2 5200+ with one physical core deactivated, fixed to 2.7GHz. The cycles achieved on this machine can obviously not be compared to the ones that can be expected on an actual sensor node, but if the cycle numbers achieved on the Athlon are already far too high, successful implementation on a sensor node seems unlikely.

Table 1 gives a comparison of the cycle counts using the unmodified Gennaro, Rabin, and Rabin (GRR) protocol, and the modification presented in (Lory, 2007) for different bitlengths and players. The results are consistent with the theory, whereby the achieved gain lowers with increased bitlength.

Table 1: Comparison of cycle counts, using the method from (Lory, 2007).

# of Bits	*n*	GRR	Lory1	Gain Lory1
160	5	5,420	4,602	15%
160	7	7,507	6,461	14%
160	9	10,051	8,190	19%
160	33	38,331	29,637	23%
160	129	190,935	152,109	20%
256	5	6,717	5,764	14%
256	7	9,504	7,981	16%
256	9	12,470	10,444	16%
256	33	47,514	38,250	19%
256	129	241,400	199,690	17%
1024	5	26,095	24,898	5%
1024	7	38,836	36,575	6%
1024	9	51,736	48,447	6%
1024	33	205,287	190,097	7%
1024	129	917,863	848,298	8%

Table 2 compares the unmodified GRR protocol with the version utilizing the optimizations presented in (Lory, 2009). The improvement is much more pronounced than when only using the method from (Lory, 2007), although, also according to the complexity theoretic computations, with increasing number of participants the gain gets lower and eventually even turns into a loss. The absolute cycle numbers here are promising, 5,000 cycles for a computation over a 256-bit field for 7 players (i.e. an attacker would have to capture and analyze 7 nodes to successfully extract the secret) hint for an at least acceptable time when implemented on a sensor node.

[1] http://gmplib.org/

Table 2: Comparison of cycle counts, using the method from (Lory, 2009).

# of Bits	*n*	GRR	Lory2	Gain Lory2
160	5	5,420	3,095	43%
160	7	7,507	5,080	32%
160	9	10,051	7,332	27%
160	33	38,331	61,798	-61%
160	129	190,935	813,058	-326%
256	5	6,717	3,282	51%
256	7	9,504	5,328	44%
256	9	12,470	7,861	37%
256	33	47,514	66,759	-41%
256	129	241,400	883,241	-266%
1024	5	26,095	4,787	82%
1024	7	38,836	7,582	80%
1024	9	51,736	11,012	79%
1024	33	205,287	95,900	53%
1024	129	917,863	1,252,135	-36%

4 OUTLOOK

Our next steps will be to exchange the GMP library, which is far too big for an efficient use on sensor nodes, with our own, custom built library, tailor made for the requirements of constraint devices. If the results are still in an acceptable range, we will port the algorithms to an actual sensor node and examine the performance in this environment.

Finally, if it turns out that we still get reasonable cycle counts, we will try incorporating the multiparty computation approach into a dedicated security protocol for wireless sensor networks.

In addition to the aforementioned work, more detailed analysis and a breakdown of the different stages of the algorithm is planned, together with more in depth comparison of the relationships between bitlength, number of players and cycle count.

ACKNOWLEDGEMENTS

Manuel Koschuch, Matthias Hudler, and Michael Krüger are supported by the MA27 - EU-Strategie und Wirtschaftsentwicklung - in the course of the funding programme “Stiftungsprofessuren und Kompetenzteams für die Wiener Fachhochschul-Ausbildungen”. Peter Lory is supported by the European Regional Development Fund - Europäischer Fonds für regionale Entwicklung (EFRE).

REFERENCES

Cramer, R. and Damgård, I. (2005). Multiparty computation, an introduction. In Catalano, D., Cramer, R., Damgård, I., Di Crescenco, G., Pointcheval, D., and Takagi, T., editors, *Contemporary Cryptology*, Advanced Courses in Mathematics CRM Barcelona, pages 41–87. Birkhäuser, Basel.

Gennaro, R., Rabin, M. O., and Rabin, T. (1998). Simplified VSS and fast-track multiparty computations with applications to threshold cryptography. In *Proceedings of the 17th ACM Symposium on Principles of Distributed Computing (PODC'98)*.

Knuth, D. (1998). *The Art of Computer Programming*, volume 2. Addison-Wesley, Reading.

Lory, P. (2007). Reducing the complexity in the distributed multiplication protocol of two polynomially shared values. In *Proceedings of the 21st International Conference on Advanced Information Networking and Applications (AINA'2007)*, volume 1, pages 404–408. IEEE Computer Society.

Lory, P. (2009). Secure distributed multiplication of two polynomially shared values: Enhancing the efficiency of the protocol. In *Proceedings of the Third International Conference on Emerging Security Information, Systems and Technologies (SECURWARE 2009)*, pages 486–491. IEEE Computer Society.

Merwe, J. V. D., Dawoud, D., and McDonald, S. (2007). A survey on peer-to-peer key management for mobile ad hoc networks. *ACM Computing Surveys (CSUR)*, 39(1):1–45.

Shamir, A. (1979). How to share a secret. *Communications of the ACM*, 22(11):612–613.

Wenzl, J. (2010). Laufzeitanalyse dreier Versionen eines Mehrparteien-Multiplikationsprotokolls. Regensburger Diskussionsbeiträge zur Wirtschaftswissenschaft 440, Institut für Wirtschaftsinformatik, Universität Regensburg.