OPTIMIZING CRYPTOGRAPHIC THRESHOLD SCHEMES FOR THE USE IN 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 multiplication protocol of Gennaro, Rabin and Rabin.

Abstract: A huge number of small, computationally restricted sensor nodes can be connected wirelessly to form a sensor network. Such networks can be used to monitor large areas and communicate a multitude of measurements (like temperature, humidity, radiation, and so on) to a remote base station. Since this communication happens over the air interface, the transmitted messages are susceptible to forgery, manipulation and eavesdropping. Conventional cryptographic countermeasures against these kind of attacks cannot be readily applied in the context of sensor networks, due to the limited resources of the individual nodes. Since single nodes can be very easily captured and examined, symmetric schemes with the secret key present in every (or at least a subset of) node(s) pose quite a risk in this setting. In this work, we examine the applicability of threshold cryptographic techniques, especially the Gennaro-Rabin-Rabin multiparty multiplication protocol, for sensor networks by employing several optimizations to the different steps of this algorithm, building on previous results we obtained. We are able to improve the running time up to a factor of 6 compared to an unoptimized version for a bitlength of 1,024 Bit and 33 players.

1 INTRODUCTION

Wireless Sensor Networks (WSNs), where a (potentially huge) number of small, resource-constrained sensor nodes is deployed in a large area to measure a wide variety of parameters and communicate them wirelessly in a hop-to-hop manner to a base station for evaluation are still an emerging field of technology. They can be used to efficiently monitor things like water quality, temperature distribution or radioactive particles in areas where approaches using wired devices are too costly or even impossible.

The current challenge when dealing with wireless sensor networks is the difficulty to achieve confidential, authenticated communication over the air interface. Common techniques against eavesdropping, message forgery and manipulation that can easily be deployed on stationary PCs usually do not work in WSNs, due to the huge constraints in terms of available memory, computing power and energy of the individual nodes. The usual way to secure WSNs today is to use symmetric cryptographic techniques, which in general can be calculated much more efficiently than their asymmetric counterparts. The problem with this approach is the storage and distribution of the keys: two sensor nodes can only communicate when they share a common symmetric key. But due to the special structure of WSNs, the loss or malicious removal of single nodes goes largely undetected, so that an attacker can easily try to extract the secret key from a captured node. To avoid the whole network becoming compromised by such an attack, usually only a

Koschuch M., Hudler M., Krüger M., Lory P. and Wenzl J..
OPTIMIZING CRYPTOGRAPHIC THRESHOLD SCHEMES FOR THE USE IN WIRELESS SENSOR NETWORKS - Position Paper.
DOI: 10.5220/0003607400750078
In *Proceedings of the International Conference on Data Communication Networking and Optical Communication System* (DCNET-2011), pages 75-78
ISBN: 978-989-8425-69-0

certain number of nodes share the same key, which raises the new problem of key distribution. A general overview of different key management techniques usable in WSNs is given in (Merwe et al., 2007).

Given these special challenges, the use of threshold cryptography becomes attractive: instead of storing the secret key on a single node, a number $t+1$ of uncompromised nodes must cooperate to generate a valid secret. Capturing a single node is now useless for an attacker, he has to gain access to at least *t+1* nodes to extract the individual shares of the secret and combine them. There is a multitude of threshold cryptography schemes proposed in the literature, their main problem usually being the computational complexity.

In this work we extend our previous work on the subject by further trying to optimize the Gennaro, Rabin and Rabin (GRR)(Gennaro et al., 1998) protocol, thereby improving the applicability of this protocol in the context of sensor nodes.

The remainder of this position paper is structured as follows: Section 2 gives a general introduction to the protocol of Gennaro, Rabin and Rabin and details of our optimizations. Section 3 then presents our current experimental results, while finally Section 4 gives an outlook on our next steps planned.

2 THE PROTOCOL OF GENNARO, RABIN AND RABIN

Classical theoretical results (Ben-Or et al., 1988; Chaum et al., 1988; Goldreich et al., 1987; Yao, 1986) show that any multiparty computation can be performed securely if the number of corrupted participants does not exceed certain bounds. For a survey of these results the reader is referred to the article of Cramer and Damgård (Cramer and Damgård, 2005).

Unfortunately, without further optimizations these results are not easily applicable in real world applications. One of the most prominent examples for the efforts to accelerate these approaches is the paper of Gennaro, Rabin and Rabin (Gennaro et al., 1998). Among other results, it presents a more efficient variant of the Ben-Or, Goldwasser and Wigderson (Ben-Or et al., 1988) multiplication protocol. It gives a protocol for the fast multiparty multiplication of two polynomially shared values over $\mathbb{Z}_q$ with a public prime number q.

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). 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 $\alpha\beta = H(0)$. Multiplication protocols of this type are important cryptographic primitives. In particular, they play a decisive role in comparing shared numbers (Damgård et al., 2006) and in the shared generation of an RSA modulus by a number of participants such that none of them knows the factorization (Algesheimer et al., 2002; Catalano, 2005).

The multiplication protocol of Gennaro, Rabin and Rabin (Gennaro et al., 1998) consists of two steps and requires one round of communication and $O(n^2k\log n + nk^2)$ bit-operations per player, where k is the bit size of the prime q and n is the number of players.

In step 1. player P_i $(1 \leq i \leq 2t+1)$ computes $f_\alpha(i)f_\beta(i)$ and shares this value by choosing a random polynomial $h_i(x)$ of maximum degree t, such that $h_i(0) = f_\alpha(i)f_\beta(i)$. He then gives player P_j $(1 \leq j \leq n)$ the value $h_i(j)$.

In (Lory, 2007) a modification of this step is given, which reduces its complexity from $O(n^2k\log n)$ to $O(n^2k)$ (and thus the complexity of the entire protocol to $O(n^2k+nk^2)$) by utilization of Newton's scheme of divided differences.

However, in many practical situations (e. g. the above mentioned shared generation of an RSA modulus) k (typically $k = 1024$) will exceed n and the $O(nk^2)$-term will still dominate. For these cases, in (Lory, 2009) a protocol is given, which modifies step 2 to require only $O(n^2k)$ bit-operations per player. All of the above mentioned optimizations were also implemented and subsumed in (Koschuch et al., 2010).

In this work, we perform an additional investigation of step 2: in this step, each player $P_j(1 \leq j \leq n)$ determines his share $H(j)$ of $\alpha\beta$ by locally computing the linear combination

$$H(j) = \sum_{i=1}^{2t+1} \lambda_i h_i(j), \qquad (1)$$

where the values $h_i(j)$ have been communicated to him by players $P_i(1 \leq i \leq 2t+1)$ during step 1. Here, the λ_i are the coefficients of Lagrange's interpolation formula of degree $2t$, which interpolate the support abscissas $i = 1, 2, ..., 2t+1$ to 0. In general, for a polynomial of degree $d-1$ these known non-zero-

constants are given by

$$\lambda_i^{(d)} = \prod_{\substack{1\le k\le d\\ k\neq i}} \frac{k}{k-i} \bmod q. \qquad (2)$$

Expanding this equation becomes:

$$\begin{aligned}\lambda_i^{(d+1)} &= \frac{1\cdot 2\cdot\ldots\cdot(i-1)\cdot(i+1)\cdot\ldots\cdot d\cdot(d+1)}{(-(i-1))\cdot(-(i-2))\cdot\ldots\cdot(-1)\cdot 1\cdot 2\cdot\ldots\cdot(d-i)\cdot(d+1-i)}\\ &= (-1)^{i-1}\frac{(d-i+2)\cdot(d-i+3)\cdot\ldots\cdot d\cdot(d+1)}{2\cdot 3\cdot\ldots\cdot i}.\end{aligned}$$

Consequently

$$\begin{aligned}|\lambda_i^{(d+1)}| &= \frac{(d-i+2)\cdot(d-i+3)\cdot\ldots\cdot d\cdot(d+1)}{2\cdot 3\cdot\ldots\cdot i},\\ |\lambda_i^{(d)}| &= \frac{(d-i+1)\cdot(d-i+2)\cdot\ldots\cdot(d-1)\cdot d}{2\cdot 3\cdot\ldots\cdot i},\\ |\lambda_{i-1}^{(d)}| &= \frac{(d-i+2)\cdot(d-i+3)\cdot\ldots\cdot(d-1)\cdot d}{2\cdot 3\cdot\ldots\cdot(i-1)},\end{aligned}$$

and

$$\begin{aligned}|\lambda_{i-1}^{(d)}| + |\lambda_i^{(d)}| &= \frac{i\cdot(d-i+2)\cdot(d-i+3)\cdot\ldots\cdot(d-1)\cdot d}{2\cdot 3\cdot\ldots\cdot i} + \frac{(d-i+1)\cdot(d-i+2)\cdot\ldots\cdot(d-1)\cdot d}{2\cdot 3\cdot\ldots\cdot i}\\ &= \frac{(d-i+2)\cdot(d-i+3)\cdot\ldots\cdot(d-1)\cdot d}{2\cdot 3\cdot\ldots\cdot i} * \frac{(i+d-i+1)}{1}\\ &= \frac{(d-i+2)\cdot(d-i+3)\cdot\ldots\cdot(d-1)\cdot d\cdot(d+1)}{2\cdot 3\cdot\ldots\cdot i}\\ &= |\lambda_i^{(d+1)}|.\end{aligned}$$

From this it follows that for equidistant support abscissas $i = 1, 2, \ldots, d$ (as they are used in the GRR protocol) the unreduced coefficients $\lambda_i^{(d)}$ of Lagrange's interpolation formula of degree $d-1$ obey the recursion

$$|\lambda_i^{(d+1)}| = |\lambda_{i-1}^{(d)}| + |\lambda_i^{(d)}| \qquad (3)$$

This and trivial initial values demonstrate that the $\lambda_i^{(d)}$ are always integers.

This fact has the consequence that the reduced coefficients as given by Equation (2) can be calculated very easily, because no computation of a modular inverse is necessary. In order to keep the absolute values of the coefficients low, the reduction should not be done into $\mathbb{Z}_q = \{x \in \mathbb{Z} | 0 \le x < q\}$. Rather, the coefficients should be from $z_q := \{x \in \mathbb{Z} | -q/2 < x \le q/2\}$ (Algesheimer et al., 2002). For small values of $d = 2t+1$ this guarantees small absolute values for the coefficients and saves computing time.

3 PRELIMINARY RESULTS

Tables 1 and 2 give the comparison between step 2 of the unmodified GRR protocol with the modifications made in (Lory, 2009) and in this work, respectively. The first version is the straightforward implementation of the unoptimized GRR protocol, with coefficients λ_i in the interval $\mathbb{Z}_q$; the second version is designed for small values of n as presented in (Lory, 2009); the third version finally exploits the observations of this work and uses coefficients λ_i from z_q. All the computations use the GNU multiple precision arithmetic library [1] in version 5.0.1 and are on an AMD Athlon64 X2 5200+ with one physical core deactivated, fixed to 1.0GHz. The results obtained on this setup can obviously not be compared to those achievable on actual sensor hardware, but if the cycle count on this test setup is already far too large, the proposed solution obviously does not work as expected.

Table 1: Comparison of the running time in milliseconds of step 2 of the unmodified GRR protocol and our optimizations of this protocol, as published in (Koschuch et al., 2010). k denotes the bitlength, n the number of players.

$k = 1024$	GRR	(Koschuch et al., 2010)
$n = 5$	0.047	0.018
$n = 9$	0.154	0.081
$n = 33$	2.218	2.866
$n = 129$	40.495	154.847

Table 2: Comparison of the running time in milliseconds of step 2 of the unmodified GRR protocol and the additional optimizations of this protocol from this work. k denotes the bitlength, n the number of players.

$k = 1024$	GRR	Reduction to z_q
$n = 5$	0.047	0.009
$n = 9$	0.154	0.027
$n = 33$	2.218	0.371
$n = 129$	40.495	12.588

Tables 1 and 2 show the comparison of step 2 of the unmodified protocol with the optimizations detailed in (Lory, 2009) and the ones performed in this work, respectively. Our new approach with reduction to z_q improves the running times significantly, up to a factor of 6 when compared to an unmodified GRR implementation. In addition, it can be assumed that this reduction also results in significantly less memory requirements during protocol execution, although this still remains to be proven by complementary measurements.

4 OUTLOOK

Our preliminary results look promising and clearly indicate an additional performance improvement when using the optimizations proposed in this work. The

[1] http://gmplib.org

next steps will be to replace the GMP library with our own code, optimized for constrained devices and much smaller than the GNU library and finally porting the protocol to a sensor node to get the timings on real hardware.

In addition, we also plan to perform a more detailed analysis of the algorithm, including several different bitlengths and numbers of players.

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

Algesheimer, J., Camenisch, J., and Shoup, V. (2002). Efficient computation modulo a shared secret with application to the generation of shared safe-prime products. In Yung, M., editor, *Advances in Cryptology – CRYPTO 2002*, number 2442 in Lecture Notes in Computer Science, pages 417–432. Springer Berlin.

Ben-Or, M., Goldwasser, S., and Wigderson, A. (1988). Completeness theorems for non-cryptographic fault-tolerant distributed computation. In *Proceedings of the 20th Annual Symposium on Theory of Computing (STOC'88)*, pages 1–10. ACM Press.

Catalano, D. (2005). *Contemporary Cryptology, Advanced Courses in Mathematics - CRM Barcelona*, chapter Efficient distributed computation modulo a shared secret, pages 1–39. Birkhäuser, Basel.

Chaum, D., Crépeau, C., and Damgård, I. (1988). Multiparty unconditionally secure protocols. In *Proceedings of the 20th Annual Symposium on Theory of Computing (STOC'88)*, pages 11–19. ACM Press.

Cramer, R. and Damgård, I. (2005). *Contemporary Cryptology, Advanced Courses in Mathematics - CRM Barcelona*, chapter Multiparty computation, an introduction, pages 41–87. Birkhäuser, Basel.

Damgård, I., Fitzi, M., Kiltz, E., Nielsen, J., and Toft, T. (2006). Unconditionally secure constant-rounds multi-party computation for equality, comparison, bits and exponentation. In *Proceedings of the 3rd Theory of Cryptography Conference (TCC'2006)*, number 3876 in Lecture Notes in Computer Science, pages 285–304. Springer Berlin.

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)*.

Goldreich, O., Micali, S., and Wigderson, A. (1987). How to play any mental game. In *Proceedings of the 19th Annual Symposium on Theory of Computing (STOC'87)*, pages 218–229. ACM Press.

Koschuch, M., Hudler, M., Krüger, M., Lory, P., and Wenzl, J. (2010). Applicability of multiparty computation schemes for wireless sensor networks - position paper. In Sevillano, J. L., Obaidat, M. S., and Nicopolitidis, P., editors, *DCNET 2010 - International Conference on Data Communication Networking - Proceedings of DCNET and OPTICS 2010*, pages 125–128. SciTePress - Science and Technology Publications.

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.

Yao, A. C. (1986). How to generate and exchange secrets. In *Proceedings of the 27th IEEE Symposium on Foundations of Computer Science (FOCS'86)*, pages 162–167. IEEE Computer Society.