Information Security for Sensors by Overwhelming Random Sequences and Permutations *

Shlomi Dolev
Department of Computer Science
Ben-Gurion University
Beer-Sheva, 84105, Israel
dolev@cs.bgu.ac.il

Niv Gilboa
Department of Computer Science
Ben-Gurion University
Beer-Sheva, 84105, Israel
niv.gilboa@gmail.com

Marina Kopeetsky
Department of Software Engineering
Sami-Shamoon College of Engineering
Beer-Sheva, 84100, Israel
marinako@sce.ac.il

Giuseppe Persiano
Dipartimento di Informatica ed Applicazioni
Università di Salerno
Via Ponte Don Melillo Salerno
84084 Campania Italy
giuper@dia.unisa.it

Paul G. Spirakis
Department of Computer Engineering and Informatics
University of Patras and Research Academic Computer Technology Institute
N. Kazantzakis str., University Campus, 265 00 Rio, Patras, Greece
spirakis@cti.gr

ABSTRACT

We propose efficient schemes for information-theoretically secure key exchange in the Bounded Storage Model (BSM), where the adversary is assumed to have limited storage. Our schemes generate a secret One Time Pad (OTP) shared by the sender and the receiver, from a large number of public random bits produced by the sender or by an external source. Our schemes initially generate a small number of shared secret bits, using known techniques. We introduce a new method to expand a small number of shared bits to a much longer, shared key.

Our schemes are tailored to the requirements of sensor nodes and wireless networks. They are simple, efficient to implement and take advantage of the fact that practical wireless protocols transmit data in frames, unlike previous protocols, which assume access to specific bits in a stream of data. Indeed, our main contribution is twofold.

On the one hand, we construct schemes that are attractive in terms of simplicity, computational complexity, number of bits read from the shared random source and expansion factor of the initial key to the final shared key.

On the other hand, we show how to transform any existing scheme for key exchange in BSM into a more efficient scheme in the number of bits it reads from the shared source, given that the source is transmitted in frames.

*An extended abstract of this paper was submitted to DIALM-POMC 2010. Partially supported by the ICT Programme of the European Union under contract number FP7-215270 (FRONTS), Microsoft, NSF, Deutsche Telekom, Rita Altura Trust Chair in Computer Sciences, Lynne and William Frankel Center for Computer Sciences, and the internal research program of Sami Shamoon College.

CCS'10, October 4–8, 2010, Chicago, Illinois, USA.
ACM 978-1-4503-0244-9/10/10.

Categories and Subject Descriptors

C.2.0 [**COMPUTER-COMMUNICATION NETWORKS**]: General—*Security and protection*

General Terms

Security

Keywords

Information Theoretic Security, Bounded Storage Model,Wireless Network

1. INTRODUCTION

A major building block in security and cryptography is generation of a secret that two parties share. The secret may then be used as a symmetric encryption or authentication key.

We propose a scheme to generate a shared key in the BSM. The BSM was presented in Maurer's work [7]. This model investigates cryptographic tasks such as encryption and authentication in the presence of an adversary that has bounded storage capacity. While most of modern cryptography limits an adversary's resources, the usual approach is to place a bound on the adversary's time complexity. Given various unproven assumptions on the hardness of computational tasks, modern cryptography has many beautiful constructions of schemes that are secure against an adversary that has limited time complexity.

In the Bounded Storage Model, on the other hand, there is no need for computational assumptions. Given a source of random bits that broadcasts more traffic than the adversary can store, legitimate parties can perform cryptographic tasks in a way that is information-theoretically secure. This is true even if the storage of the legitimate parties is smaller than that of the adversary.

One of the main cryptographic tasks is for two parties to share a key, without leaking any of its bits to an adversary that monitors traffic. [7] showed that a key can be shared even when the two

parties do not share any bits before the protocol begins. This work was improved by [3], and this second work was analyzed in [5] and shown to be essentially optimal in terms of the amount of data the two parties can share, given the ratio between the storage capacity of the adversary and the storage capacity of the two legitimate parties.

Subsequent works [1], [2], [4], [6] and [9] showed schemes to expand a small initial key to a much larger key that can be used as a OTP. Both the initial key and the OTP are shared by the legitimate parties, but are unknown to the adversary. It is assumed that the adversary has no information on the initial key with probability 1, while the probability that it has some information on the one-time pad is less than some parameter ϵ.

Our contribution.

We propose a pair of two-stage schemes that first use the process for initial key generation of [3] to generate a short, shared key. Our schemes then employ a novel method for expanding a short initial key into a longer key. Our scheme has the basic property of key exchange schemes that passive attackers, who only monitor traffic, do not obtain information on the shared key, while active attackers may mount Man-in-the-Middle attacks. We may assume that such attacks are foiled by identification performed in the physical layer (e.g., [10]).

The basic step of our schemes is to use the initial key for both the sender and the receiver to select several blocks of bits from the shared random source. After all the random bits have been transmitted, the sender chooses a random permutation on all the stored bits and exchanges it with the receiver. After permuting the bits, both parties exclusive-or all the bits in a contiguous block of bits, thus obtaining a single bit of the OTP. Given enough such blocks, they construct the whole OTP.

We present two protocols the *Permutation Revealing Protocol* PRP and *Permutation Encrypted Protocol* PEP. The permutation in PRP is sent as clear text, while in PEP it is kept secret forever. Thus, PEP may be used an exponential number of times. Each time, a new string of n random bits must be transmitted for the sender and receiver to generate a new OTP of m bits.

We use the following notation: k denotes the security parameter which means that all schemes are information-theoretically secure with probability at least $1-\epsilon = 1-2^{-k}$. The length of the OTP is m and the length of the random string is denoted by n. In our work, $n = mb(k + \log m)$ for a parameter b, which we call the number of channels. Finally, the frame length for wireless communication is α bits.

The complexity of PRP under various measures is as follows. The computational complexity is $m(k + \log m)$. The number of bits read from the random source is $\lceil \frac{m}{\alpha} \rceil \alpha(\log m + k)$. The expansion factor, which is defined as the ratio between the initial secret (the product of the first stage of the protocol) and the OTP length m is $\frac{m}{\log b(\log m+k)}$. The storage required for the second stage of PRP is $O(m(k + \log m))$. The storage required for the first stage is identical to the storage of [3].

A second contribution is to transform a key exchange scheme that accesses distinct bits in the random strings into a scheme that accesses blocks of α bits (where each block is identified with a frame of the wireless protocol). We can thus reduce the number of bits that a scheme reads. Applied to Vadhan's scheme [9], which reads the least number of bits of all known schemes, we obtain a scheme that reads $k + \log m$ bits (compared to $k + \log n$).

2. SETTING AND NOTATION

Consider a *wireless network* which consists of n nodes. A sender, S, wishes to send information securely to a receiver, R. S intends to encrypt its message in blocks of m bits. Each block is encrypted by a one-time pad of length m bits. S and R perform a key exchange scheme to share an m-bit one-time pad prior to sending an encrypted block.

We assume a bounded storage model in which all wireless nodes have the same storage capacity sp, while an adversary has capacity s_{Ad} such that possibly $s_{Ad} > sp$. S shares m bits with R by generating and sending T random bits, $sp < s_{Ad} < T$.

The T bits are sent over b channels, which may have different physical implementations such as different frequencies or different time slots on the same frequency. We denote the channels by $c_1, c_2, \ \ldots, c_b$.

The sender node S and the receiver node R simultaneously run two independent processes in order to generate the shared OTP.

Process 1 runs in the background continuously; its purpose is to generate (a small number of) shared random bits by using the scheme of [3] as follows. S transmits to R a random string of length T bits. In order to generate one secret shared bit, S and R randomly record $O(\sqrt{T})$ bits and their locations, using $O(\sqrt{T}\log T)$ bits of memory. Then, S and R send each other the locations of the stored bits, without revealing the actual values of these bits. Due to the Birthday paradox [3], with high probability there is at least one shared location for S and R. Assuming that T is significantly larger than s_{Ad}, there is high probability that the adversary does not know this shared bit. Standard techniques may use repetitions to make the probability that the bit is unknown to the adversary as close to 1 as necessary.

The shared secret bits produced by *Process 1* are expensive it terms of the time (and number of non-shared random bits produced by the sender) needed to produce a secret shared bit. *Process 2* expands this computationally expensive random string and derives a much longer OTP.

3. PERMUTATION REVEALING PROTOCOL

The input of PRP is a set of channels, $c_1, \ldots, c_b$, a security parameter k and the required length m of a shared OTP, which is the output of PRP . As previously stated, PRP has two processes. The first process begins without any shared random bits and generates a small shared secret for R and S. This shared secret, of length $\log b(\log m + k)$ is regarded as $\log m + k$ indices. Each index determines one of the b channels.

Process 2 is performed in two phases. During the first phase, S sends to R a large number of random bits over the wireless network. This traffic is transmitted in frames of size α bits per frame. S and R use the small key they share to determine which of the frames that S sends in the first phase must be received and stored. The product of the first phase is a large number of shared bits for S and R. In the second phase, S and R combine subsets of their shared bits to derive an m-bit shared key. The main point of PRP is that the adversary does not have enough space to store all the random bits of phase 1. The combination of bits in phase 2 make it very likely that for every bit in the OTP, the adversary misses at least one of the bits that generate it, and thus the adversary has no information on any of the OTPbits.

Diving into the details we note that S transmits random data over b different channels in phase 1 of *Process 2*. Let $\lambda = \log m + k$. The data in each channel is logically organized in λ blocks of m bits each (while each such block may physically include several frames of α bits). Thus for every $j = 1, \ldots, \lambda$ there are b different blocks of m bits (one on each channel). The shared key s defines the correct channel s_j for block j. R receives the next $\lceil m/\alpha \rceil$ frames from channel s_j and thus obtains a block of m bits.

In phase 1, S sends $bm\lambda$ bits. Both S and R store only $m\lambda$ bits, denoted by $R_{1,s_1}, \ldots, R_{\lambda,s_\lambda}$.

In the second phase of PRP , S sends to R in clear text (possibly monitored by the adversary), a random permutation π. This permutation defines a reordering of the bits of the concatenated shared string $R_{1,s_1} \| \ldots \| R_{\lambda,s_\lambda}$. In order to determine π, S has to generate and send $m\lambda \log(m\lambda)$ random bits. Here $m\lambda$ bits is the number of bits in the concatenated shared string, and $\log(m\lambda)$ is the number of bits needed to encode an index in this concatenated string.

Upon reception of π, R permutes the $m\lambda$ random bits received during the first phase, and generates a matrix P of λ rows and m columns. The $i-th$ bit of OTP is computed as an exclusive-or of all bits of the $i-th$ column.

We state and prove that PRP is an information-theoretically secure key exchange scheme in the Bounded Storage Model. Namely, PRP outputs an OTP of m bits that S and R share, and the probability that the adversary determines even a single bit of the OTP correctly is less than $1/2 + 2^{-k}$.

4. PERMUTATION ENCRYPTED PROTOCOL

In PEP, the number of bits shared in *Process 1*, is larger than in the PRP case. The shared key is reusable for an exponential (in the security parameter k) number of encryptions.

PEP is similar to PRP, but instead of a permutation revealing phase, the shared bits of *Process 1* define the permutation π that is used in *Process 2*. The same permutation is used over and over in N rounds to generate successive blocks of m bits for the OTP.

Here we use λ_N to denote $\log(mN) + k$. Thus, the notation λ used in the PRP section can be written as λ_1.

The length of the shared key after *Process 1* of PEP is equal to

$$\lambda_N \log b + m\lambda_N \log(m\lambda_N).$$

The first summand, $\lambda_N \log b$, defines λ_N blocks of $\log b$ bits. Each such block determines a channel on which to receive a block of bits $R_{j,i}$ (similarly to PRP).

The second summand, $m\lambda_N \log(m\lambda_N)$, determines a permutation on all the $m\lambda_N$ bits that R and S share in order to obtain m bits for an OTP.

S and R perform a similar procedure to *Process 2* in PRP to derive m bits. In phase 1, S sends $m\lambda_N$ random bits to R over each of the b channels. For each block of m bits, only a single channel is correct, while all other channels carry random dummy bits. Similarly to PRP, the correct channel is defined by bits shared in *Process 1*.

In phase 2, S and R use their shared permutation to reorder the $m\lambda_N \log(m\lambda_N)$ shared bits in a matrix of size $\lambda_N \times m$. An exclusive-or on all the bits of a matrix column yields an OTP bit. Performing this process on each of the m columns of the matrix produces an OTP of length m bits.

S and R repeat this process N times. Each time S sends new random bits and both S and R use the same permutation.

We state and prove that PEP outputs in BSM an OTP of mN bits that S and R share, and the probability that the adversary obtains even a single bit of the OTP is less than 2^{-k}.

The expansion factor of PEP can be often improved by the following procedure, which we refer to as *improved* PEP.

1. Begin with a shared, initial key of length $\log b(\log \xi + k + 1)$, where ξ is the length of the initial key for PEP (with security parameter $k+1$).
2. Use PRP to expand this key to a shared output string of length ξ.
3. Perform PEP with a shared key of length ξ.

The expansion factor of improved PEP is better than that of PEP when $\xi \geq \log b(\log \xi + k)$.

5. REFERENCES

[1] Y. Aumann, Y. Z. Ding, M. O. Rabin, "Everlasting Security in the Bounded Storage Model", *IEEE Transactions on Information Theory*, Vol. 48, No. 6, pp. 1668-1680, June 2002.

[2] Y. Z. Ding, M. O. Rabin, "Hyperencryption and Everlasting Security", *Annual Symposium on Theoretical Aspects of Computer Science (STACS)*, pp. 1-26, 2002.

[3] C. Cachin, U. Maurer, "Unconditional Security Against Memory-Bounded Adversaries", CRYPTO'97, pp. 292-306, 1997.

[4] S. Dziembowski, U. Maurer, "Tight security proofs for the bounded-storage model",34th Annual ACM Symposium on Theory of Computing (STOC'02), pp. 341-350, 2002.

[5] S. Dziembowski, U. Maurer "On Generating the Initial Key in the Bounded-Storage Model", *Advances in Cryptology-EUROCRYPT 2004*, Vol. 3027, pp. 126-137, 2004.

[6] Chi-Jen Lu, "Encryption against Storage-Bounded Adversaries from On-Line Strong Extractors", *Journal of Cryptology*, Vol. 17 No. 1, pp. 27-42, 2004.

[7] U. Maurer, "Conditionally-Perfect Secrecy and a Provably-Secure Randomized Cypher", *Journal on Cryptology*, Vol. 5, No. 1, pp. 53-66, 1992.

[8] D. R. Stinson, *"Cryptography. Theory and Practice"*, Chapman and Hall/CRC, Third edition, 2006.

[9] S. P. Vadhan, "Constructing Locally Computable Extractors and Cryptosystems in the Bounded-Storage Model", *Journal of Cryptology*, Vol. 17 No. 1, pp. 43-77, 2004.

[10] L. Xiao, L. Greenstein, N. Mandayam, and W. Trappe,"Fingerprints in the ether: Using the physical layer for wireless authentication," *Proc. IEEE International Conference on Communications*, Glasgow, Scotland, June 2007.