SAND2007-4543C

Distance-Avoiding Sets for Extremely Low-Bandwidth Authentication – 1569059341

Michael J. Collins and Scott A. Mitchell
Sandia National Laboratories*
Albuquerque, NM 87185
Email: mjcolli@sandia.gov

ABSTRACT

We develop a scheme for providing strong cryptographic authentication on a stream of messages which consumes very little bandwidth and is robust in the presence of dropped messages. Such a scheme should be useful for extremely low-power, low-bandwidth ad-hoc wireless sensor networks. The tradeoffs among security, memory, bandwidth, and tolerance for missing messages give rise to several new optimization problems. We report on experimental results and derive bounds on the performance of the scheme.

1. INTRODUCTION

We consider the following scenario: we wish to send a stream of many short messages $m_1, m_2, m_3, \cdots$ on a channel with very limited bandwidth, and we wish to provide strong cryptographic authentication for this data. Because bandwidth is so limited, we assume that we must use almost all transmitted bits for delivering payload data: say we can append no more than r bits of authentication to each message, where r is too small to provide adequate security. Suppose we have decided that qr authentication bits are needed for security; a simple solution would be to send q consecutive messages $m_1, m_2, \cdots m_q$, followed by a message authentication tag t of length qr for the concatenated message $(m_1|m_2|\cdots m_q)$ (repeating this process for the next block of q messages and so on). This achieves the desired data rate, but it is unsatisfactory for several reasons. In an extremely low-power environment (such as a wireless network of very small sensors), we expect that many messages will be dropped or corrupted, making it impossible for the receiver to verify the correctness of t. Also, we are transmitting no data at all during the relatively long time needed to transmit the tag. We seek a more robust solution which will tolerate some missing messages (without the additional cost of applying an error-correction scheme to already-redundant data), and which does not interrupt the flow of payload data.

2. SUBSET AUTHENTICATION

Our basic approach is to append a short authentication tag a_i to each message m_i; each a_i is an r-bit authentication tag for some appropriately-chosen subset S_i of the previous messages. Let $\mathcal{A}_K$ be a message authentication code (MAC) with key K that produces an r-bit output. Thus if $S_i = \{j_1^i < j_2^i < \cdots j_k^i\}$ we have[1]

$$a_i = \mathcal{A}_K(i|m_{j_1^i}|\cdots|m_{j_k^i}) \tag{1}$$

and we transmit $m_1, a_1, m_2, a_2, \cdots$. If each message is contained in q sets, then each message is used in the computation of q different tags, and we will eventually accumulate the required qr bits of authentication for each message. If $\mathcal{A}_K$ is a pseudorandom function, an adversary cannot cause an invalid m_i to be accepted without guessing qr random bits. In practice, $\mathcal{A}_K$ could be implemented by truncating the output of a full-length authentication code such as HMAC [1]. The design of a secure, less computationally intensive MAC which inherently produces a short output would pose an interesting and challenging problem.

However, it is not enough to simply require that each message appear q times. We are assuming a very low-power network with no acknowledgement or retransmission protocol, no error-correction mechanism, and occasional loss of connectivity. Thus we must expect that some messages will be lost, and if m_j is lost, all tags a_i such that $j \in S_i$ will be useless. Therefore every message must be contained in more than q sets, to provide robustness against the expected missing messages. The question is, what conditions must we impose on the sets S_i, and what is the optimal way to achieve those conditions?

We first consider the following requirement (more general requirements are considered in section 3): if any one message is lost, this must not prevent full authentication of any other message. This means that for any pair of messages m_i, m_j, we must have at least q sets which contain m_i but not m_j. Thus if m_j is lost,

*Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

[1] It is convenient to ignore the distinction between a message and its index, writing $j \in S_i$ instead of $m_j \in S_i$.

we still have enough good tags to authenticate m_i with the desired degree of security.

This "set-cover" approach requires the sender to remember many old messages. If a node can remember at most v old messages, then we must have $S_i \subset [i-v, i]$ for all i. Memory is presumably quite limited since we are dealing with very low-power nodes. Note that v is also the maximum delay before a message finally achieves full authentication, which is another reason to limit v.

Thus we have the following problem: Given memory bound v, find sets S_i that maximize q where

- For each $i \in \mathbb{N}$, $S_i \subset [i - v, i]$
- For each $i \neq j$ there are at least q sets S with $i \in S$, $j \notin S$

Given a collection of sets S, define the *strength* of the collection as

Definition 1.

$$q(S) = \min_{i,j} \#\{t | i \in S_t, j \notin S_t\}$$

(here $\#A$ denotes the size of a set A). We have defined S as an infinite collection; such a collection would of course be specified either by rotating through a finite collection of given sets, or by specifying a way to generate S_i as a function of S_{i-1}. To get the process started, we can implicitly have dummy messages $m_{-v}, \cdots m_{-1}, m_0 = 0$.

2.1 Sliding-Window Construction

We first consider the special case in which each set S_i is defined by a "sliding window"; we select a set of distances $\delta = \{\delta_1 < \cdots \delta_k \leq v\}$ and let each $S_i = \{i - \delta_1, \cdots i - \delta_k\}$.

It will be convenient to identify the vector of distances d with a binary sequence b of length v which is zero except on δ. Then

$$S_i = \{i - d : b_d = 1\}.$$

We may also treat b as an infinite sequence with $b_j = 0$ for j outside of the interval $[0, v - 1]$. We say that difference d is "realized (at j)" if $b_j = 1, b_{j+d} = 0$ and call the ordered pair $(j, j + d)$ a "realization of d". We define

Definition 2.

$$rel_b(d) = \#i | b_1 = 0, b_{i+d} = 0 \tag{2}$$

so $\text{rel}_b(d)$ is the number of times d is realized in b (we may drop the subscript b when the context is clear). We then define the strength of the vector b as

$$q(b) = \min_d \text{rel}_b(d) \tag{3}$$

consistent with the definition given above for $q(S)$.

We can assume with no loss of generality that $b_0 = b_{v-1} = 1$. Changing b_0 from zero to one does not destroy any realizations of any d; changing b_{v-1} from zero to one creates one new realization of d for every d, while destroying one realization of each d with $b_{v-d-1} = 1$. Thus $q(b)$ might increase and cannot decrease.

Note that we do not need to consider differences with absolute value greater than v; for such differences we clearly have $\text{rel}(d) = \sum_i b_i$, which is a trivial upper bound on all $\text{rel}(d)$. In fact we can limit our attention to positive differences:

Lemma 1. *For all d, $rel(d) = rel(-d)$*

Proof. $\text{rel}(d) - \text{rel}(-d) = \sum_i (b_i - b_{i+d}) = 0$ □

The problem of maximizing $q(b)$ for a fixed v has apparent connections to several other problems, in particular to the problems of optimal autocorrelation and side-lobe minimization; we have

$$\text{rel}(d) = \sum_i b_i(1 - b_{i+d}) = k - \sum_i b_i b_{i+d} \tag{4}$$

Letting $\hat{b}^d$ be the sequence $0^d b 0^d$ and assuming $d < k$, and denoting the aperiodic autocorrelation [2] of a finite binary sequence s by $AA(s)$ we have

Theorem 2. *For all d, $rel(d) = \frac{v+d-AA_d(\hat{b}^d)}{4}$*

Proof. By lemma 1, we have that $\text{rel}(d) = \frac{1}{2}(\text{rel}(d) + \text{rel}(-d))$, which is one-half the number of pairs $(i, i+d)$ with $b_i \neq b_{i+d}$. And $AA_d(\hat{b}^d)$ is the number of such pairs with $b_i \neq b_{i+d}$ minus the number of pairs with $b_i = b_{i+d}$. There are $v + d$ such pairs overall, thus

$$AA_d(\hat{b}^d) = v + d - 4\text{rel}(d) \tag{5}$$

□

We can bound the maximum strength of a sequence for a given memory size v as follows:

Theorem 3. *For all b of length v,*

$$q(b) \leq \frac{v+2}{3} \tag{6}$$

Proof. We in fact prove the stronger result that

$$\min(\text{rel}(1), \text{rel}(2)) \leq \frac{v+2}{3} \tag{7}$$

Let R^s_ℓ be the number of runs of $s \in \{0, 1\}$ of length ℓ. With no loss of generality we may assume that $\ell \leq 2$; in a long run of ones or zeros, the third value can be changed without decreasing $\text{rel}(1)$ or $\text{rel}(2)$. Then we have

$$v = R^0_1 + R^1_1 + 2R^0_2 + 2R^1_2 \tag{8}$$

Runs of zeros and ones alternate, and we can assume with no loss of generality that the sequence starts and ends with 1, so we also have

$$R^1_1 + R^1_2 = 1 + R^0_1 + R^0_2 \tag{9}$$

and combining these we obtain

$$v = 2(R_1^1 + R_2^1) + R_2^0 + R_2^1 - 1 \tag{10}$$

Now $\mathrm{rel}(1) = R_1^1 + R_2^1$ since this is the number of runs of ones. Furthermore, the distance 2 will fail to be realized at $b_i = 1$ if and only if this is immediately followed by a zero-run of length one; thus (using equation 9)

$$\mathrm{rel}(2) = R_1^1 + 2R_2^1 - R_1^0 = 1 + R_2^0 + R_2^1 \tag{11}$$

therefore

$$v = 2\mathrm{rel}(1) + \mathrm{rel}(2) - 2 \tag{12}$$

and the theorem follows. □

In fact, the same bound applies to any collection of sets, without the sliding-window assumption:

Theorem 4. *For any collection of sets S with memory bound v,*

$$q(S) \leq \frac{v+2}{3} \tag{13}$$

Proof. Let b^i be the binary sequence corresponding to the set S_i. Consider v consecutive sets $S_i, \cdots S_{i+v-1}$; for any distance d they must by definition contain at least $q(S)$ realizations of d at i; that is, there are least $q(S)$ sets which contain i but not $i+d$. Thus the sequences $b^i \cdots b^{i+v-1}$ contain at least $q(S)$ realizations of d, where in sequence b^{i+t} we only count a realization at bit position t. Similarly the sequences $b^{i+1} \cdots b^{i+v}$ contain at least $q(S)$ *different* realizations of d and so, for any L, the $v+L-1$ sequences $b^i \cdots b^{i+v+L-2}$ contain qL different realizations of d. Thus as L approaches infinity, the average value of $\mathrm{rel}(d)$ approaches (at least) $q(S)$. In particular this holds for $d = 1, 2$. Now from the proof of theorem 3, we know that

$$2\mathrm{rel}_{b^j}(1) + \mathrm{rel}_{b^j}(2) \leq v + 2$$

for each sequence b^j, thus the same must be true of the average, i.e.

$$3q(S) \leq v + 2$$

□

It is unknown whether the maximum strength of an arbitrary collection of sets can exceed the maximum strength achieved by a sliding window. The proof of theorem 4 shows that if this is the case, we must have a collection of sliding windows in which the average value of each $\mathrm{rel}(d)$ exceeds the maximum strength of any single sliding window.

We also have the following relationship among different distances:

Theorem 5. *For all d, d'*

$$rel(d) + rel(d') \geq rel(d + d') \tag{14}$$

In particular,

$$2rel(d) \geq rel(2d)$$

Proof. If $d \neq d'$ define a mapping from realizations of $d + d'$ to realizations of d and d' as follows: for each $b_i > b_{i+d+d'}$, map $(i, i + d + d')$ to $(i, i + d)$ if $b_{i+d} = 0$, else map to $(i+d, i+d+d')$. Clearly this map is injective.

If $d = d'$ then similarly every realization of $2d$ can be mapped to exactly one realization of d, and no more than two realizations of $2d$ can map to the same point. □

2.2 Optimal Sequences for Small Memory Bounds

For small values of v, optimal sequences can be found by exhaustive search; results are summarized in table 1. Only "critical" values are shown, i.e. v at which the maximum $q(b)$ changes. These results show that the bound of theorem 3 can be attained for small v. For all lengths except $v = 35$, the table gives the lexicographically smallest vector attaining $\max q(b)$. Exhaustive search was not completed for $v = 35$, but $q(b) = 11$ is still known to be optimal; if we had b of length 35 attaining $q(b) = 12$, we could remove one bit to obtain $q(b) = 11$ at length 34, which has been ruled out by exhaustive search.

As a secondary objective, we could seek to minimize the Hamming weight of b: this weight is the number of messages that must be combined to compute each authentication tag, so reducing this weight may reduce the amount of work needed to compute a_i. For all v in this table, the majority of optimal vectors have weight greater than $\frac{v}{2}$; for all these v (except 21 and possibly 35) there are no optimal vectors with weight less than $\frac{v}{2}$.

2.3 Lower Bounds for the Sliding Window Construction

If $v + 1$ is a power of 2 we can attain $q(b) \geq \frac{v+1}{4}$ by letting b be the output of a linear feedback shift register [3] with period v. The perfect autocorrelation of an LFSR implies that $\frac{v+1}{2}$ bit positions will have $b_i = 1$, and for any $d > 0$ exactly half of these will have $b_{i+d} = 0$ where the indices are taken modulo v.

More generally, if a difference set [4] D of size v exists, then we attain $q(b) \geq \frac{v+1}{4}$ by letting $b_i = 1$ precisely when $i \in D$. A difference set is a set of $k = \frac{v-1}{2}$ integers mod v, such that for each $d > 0$, there are exactly $\frac{k-1}{2}$ pairs $a, b \in D$ with $a - b = d$; this implies that there are exactly $\frac{k+1}{2}$ pairs $a \in D, b \notin D$ with $a - b = d$, hence $\mathrm{rel}(d) \geq \frac{k+1}{2} = \frac{v+1}{4}$.

LFSRs and difference sets are, however, periodic structures which do not take advantage of the edge effects inherent in this problem, and they do not provide optimal solutions. It appears to be possible to do somewhat better than $\frac{v+1}{4}$ for all v (see section 2.4), although it also seems that the maximum $q(b)/v$ approaches $\frac{1}{4}$ as v goes to infinity.

2.4 Iterative Improvement of Windows

v	$\max q(v)$	an optimal vector
1	1	1
4	2	1 1 0 1
7	3	1 1 0 0 1 0 1
10	4	1 1 0 1 0 1 0 0 1 1
14	5	1 1 1 0 0 1 0 1 0 1 1 0 0 1
17	6	1 1 1 0 0 1 0 1 1 0 0 1 1 0 1 0 1
21	7	1 1 1 0 0 0 1 0 1 0 1 1 0 1 0 0 1 1 0 0 1
24	8	1 1 1 0 0 0 1 0 1 1 0 1 0 0 1 1 0 0 1 1 0 1 0 1
27	9	1 1 1 0 0 1 0 1 0 1 0 1 1 0 0 1 0 1 1 0 0 0 1 1 0 1 1
31	10	1 1 1 1 0 0 0 1 1 0 1 0 1 0 0 1 1 0 0 1 1 0 1 0 0 1 1 0 1 0 1
35	11	1 1 0 1 0 1 0 0 1 0 0 1 1 1 1 0 0 1 1 0 0 0 1 0 1 1 0 0 1 0 1 0 1 1 1

Table 1: Optimal $q(b)$ for small v

v	max known $q(b)$	Min weight attaining max $q(b)$
40	12	19
60	18	30
100	28	48
200	55	100
300	81	150

Table 2: Best known $q(b)$ for large v

Starting with a random binary vector b^0, we can attempt to maximize q by iterative local improvement. At each step, we change one bit of the current solution b^i. If we can attain $q(b^{i+1}) > q(b^i)$ by flipping a single bit, we do this (note that a single bit change cannot increase the strength of the vector by more than 1, since it cannot change any $\mathrm{rel}(d)$ by more than 1). If such immediate improvement is not possible, we consider the set of distances d which are tight, i.e. which have $\mathrm{rel}(d) = q(b^i)$. The local optimization criteria is to reduce the size of this set as much as possible, subject to the condition that strength does not decrease (i.e. that there is no d for which $\mathrm{rel}(d)$ decreases to $q(b^i) - 1$). If local improvement is impossible, we flip two bits at random.

In order to implement this search, note that it is not necessary to recompute q from scratch for every vector at Hamming distance 1 from b^i. Instead, for each bit position i and for each tight distance d, we can compute in constant time the effect on $\mathrm{rel}(d)$ of flipping bit i. Table 2 gives the strengths of the best vectors found in this manner.

3. MORE GENERAL INDEPENDENCE CONDITIONS

More generally we may consider conditions of the following form: for parameters (r, r'), require that loss of any r messages does not prevent authentication of more than r' remaining messages. The problem considered above is the special case $r = 1, r' = 0$. In the general case we have the following: for any set A with $\#A = r$, there can be no more than r' indices $i \notin A$ such that

$$\#\{j | i \in S_j, A \cap S_j = \emptyset\} < q$$

This is a difficult condition to deal with in general, so we still consider some special cases, and still consider only the sliding-window approach. If we have $r = 1$ but $r' > 0$, then we no are no longer maximizing the minimum value of $\mathrm{rel}(d)$; instead we seek to maximize the $(r' + 1)^{\mathrm{th}}$ smallest value. The r' smallest values correspond to the r' messages for which we are allowed to loose full authentication.

If we have $r > 1$ and $r' = 0$, then we require that the loss of any set of r messages does not prevent authentication of any other message. For this case we define $\mathrm{rel}(d_1, d_2, \cdots d_r)$ as the number of indices j where $b_j = 1, b_{j+d_1} = \cdots = b_{j+d_r} = 0$, and maximize $q_r(b) = \min \mathrm{rel}(d)$ over all vectors d, where we may assume $i < j$ implies $d_i < d_j$ since order does not matter. Note that the d_i may be negative. Trivially we have

$$q_r(b) \leq \frac{v + r}{r + 1} \tag{15}$$

since every realization of $d = (1, 2, \cdots r)$ (except at $b_{v-1} = 1$) consists of a 1 followed by r zeros, and these cannot overlap. Table 3 gives the best known values of q_2 for various memory bounds v; in general these can be attained while simultaneously coming close to the best known $q = q_1$.

Acknowledgment

The authors would like to thank Carl Diegert for helpful discussions, and Austin McDaniel for assistance with implementing iterative search.

4. REFERENCES

v	max known $q_2(b)$	Best $q_1(b)$ for this q_2
10	2	3
20	4	6
30	5	9
40	7	12
60	10	16
100	16	26

Table 3: Best known $q_2(b)$ for some v

[1] Mihir Bellare, Ran Canetti, and Hugo Krawczyk. Message authentication using hash functions: the HMAC construction. *CryptoBytes*, 2(1):12–15, Spring 1996.

[2] Raymond A. Kristiansen. On the aperiodic autocorrelation of binary sequences. Master's thesis, University of Bergen, 2003.

[3] Rudolf Lidl and Harald Niederreiter. *Introduction to finite fields and their applications.* Cambridge University Press, New York, NY, USA, 1986.

[4] Jr. Marshall Hall. *Combinatorial theory (2nd ed.).* John Wiley & Sons, Inc., New York, NY, USA, 1998.