Distance-Avoiding Sequences for Extremely Low-Bandwidth Authentication

Michael J. Collins and Scott Mitchell

Sandia National Laboratories*
Albuquerque, NM USA 87185
mjcolli@sandia.gov

Abstract. We develop a scheme for providing strong cryptographic authentication on a stream of messages which consumes very little bandwidth (as little as one bit per message) and is robust in the presence of dropped messages. Such a scheme should be useful for extremely low-power, low-bandwidth wireless sensor networks and "smart dust" applications. 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 and Previous Work

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 need 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. Such a situation might arise for power-scavenging or energy harvesting systems, since communication is generally energy-intensive relative to computation.

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

* 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.

(without the additional cost of applying an erasure code to already-redundant data), and which does not interrupt the flow of payload data.

Perrig et al. [4, 5, 2] have considered the problem of efficient authentication for lossy data streams, but our work considers a somewhat different set of issues. Protocols such as μTESLA [5] have low overhead compared to earlier authentication methods for such streams, but "low overhead" in their context means tens of bytes; our work considers a scenario in which we may add only a few bits to each message. In order to attain such extreme bandwidth efficiency, it is necessary to constrain the problem somewhat. The μTESLA protocol specifically addresses the problem of *broadcast* in sensor networks, but our work is only applicable to point-to-point communication, because we assume that the sender and receiver share a symmetric key.

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 with SHA-1 [1]. The design of an equally secure but less computationally intensive MAC which inherently produces a short output would pose an interesting and challenging problem.

It is essential to include the sequence number i in the computation of a_i, so that an attacker cannot replay the same data with different authentication bits and attack each r bit tag separately. We are here making a non-standard security assumption; an attacker only has access to a *stateful* verification oracle. Once this oracle has answered one query for a message with sequence number i, it will not answer queries (or will always answer "no") for any message with sequence number $\leq i$. This is a plausible assumption in many cases, especially for data with a short lifetime, which is likely to be the case in our intended applications. Some non-standard assumption of this type is necessary to achieve our very strong efficiency requirements.

Given that bandwidth is very constrained, we would seek to avoid explicitly transmitting the entire sequence number with each message. In particular, if

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

messages are guaranteed to be received in order (i.e. if there is a direct radio link from sender to receiver), and if we assume that no more than 2^k consecutive messages will ever be lost, then it is only necessary to transmit the k low-order bits of i.

Of course 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 then becomes, 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 4): 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, 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.

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

(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 more generally by specifying a way to generate S_i as a function of i. To get the process started, we can implicitly have dummy messages $m_{-v}, \cdots m_{-1}, m_0 = 0$.

3 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 $S_i = \{i - \delta_1, \cdots i - \delta_k\}$. Without loss of generality we can assume $\delta_1 = 0$ and $\delta_k = v$.

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.

$$\mu_b(d) = \#\{i | b_i = 1, b_{i+d} = 0\} \tag{2}$$

so $\mu_b(d)$ 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

$$\mu(b) = \min_d \mu_b(d) \tag{3}$$

consistent with the definition given above for $\mu(S)$. Then b is a *t-distance avoiding sequence* if $\mu(b) \geq t$.

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 $\mu(b)$ 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 $\mu_d = \sum_i b_i$, which is a trivial upper bound on all μ_d. In fact we can limit our attention to positive differences:

Lemma 1. *For all d, $\mu_d = \mu_{-d}$*

Proof: $\mu_d - \mu_{-d} = \sum_i (b_i - b_{i+d}) = 0$

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

Theorem 1. *For all b of length v,*

$$\mu(b) \leq \frac{v+2}{3} \tag{4}$$

Proof: We in fact prove the stronger result that

$$\min(\mu_1, \mu_2) \leq \frac{v+2}{3} \tag{5}$$

Let R_ℓ^s 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 μ_1 or μ_2. Then we have

$$v = R_1^0 + R_1^1 + 2R_2^0 + 2R_2^1 \tag{6}$$

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

$$R_1^1 + R_2^1 = 1 + R_1^0 + R_2^0 \tag{7}$$

and combining these we obtain

$$v = 2(R_1^1 + R_2^1) + R_2^0 + R_2^1 - 1 \tag{8}$$

Now $\mu_1 = R_1^1 + R_2^1$ 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 7)

$$\mu_2 = R_1^1 + 2R_2^1 - R_1^0 = 1 + R_2^0 + R_2^1 \tag{9}$$

therefore

$$v = 2\mu_1 + \mu_2 - 2 \tag{10}$$

and the theorem follows.

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

Theorem 2. *For any collection of sets S with memory bound v,*

$$\mu(S) \leq \frac{v+2}{3} \tag{11}$$

Proof: Let b^i be the binary sequence corresponding to the set S_i, i.e. $b_t^i = 1$ if and only if $i - t \in S_i$. Consider v consecutive sets $S_i, \cdots S_{i+v-1}$. These are the only sets which can contain i; thus for any distance d, at least $\mu(S)$ of these sets contain i but not $i+d$. Thus the sequences $b^i \cdots b^{i+v-1}$ together contain at least $\mu(S)$ 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 $\mu(S)$ *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 $\mu_{b^j}(d)$ for $i \leq j \leq i + v + L - 2$ approaches (at least) $\mu(S)$. In particular this holds for $d = 1, 2$. Now from the proof of theorem 1, we know that

$$2\mu_{b^j}(1) + \mu_{b^j}(2) \leq v + 2$$

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

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

It is not known whether the strength of an arbitrary collection of sets can exceed the maximum strength achievable by a sliding window. The proof of theorem 2 shows that if this is the case, we must have a collection of sliding windows in which the average value of each μ_d exceeds the maximum strength of any single sliding window.

We also have the following relationship among different distances:

Theorem 3. *For all* d, d'

$$\mu_d + \mu_{d'} \geq \mu_{d+d'} \tag{12}$$

In particular,

$$2\mu_d \geq \mu_{2d}$$

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.

3.1 Lower Bounds for the Sliding Window Construction

To obtain lower bounds, we recall the following well-known [3] concept:

Definition 3. *A* (v, k, λ)-cyclic difference set *is a subset* $D \subset \mathbb{Z}_v$ *such that* $\#D = k$ *and, for each* $d \in \mathbb{Z}_v \backslash \{0\}$*, there are exactly* λ *pairs* $a, b \in D$ *with* $a - b = d$

If such a set exists we must have $\lambda(v-1) = k(k-1)$. In particular, if $k = \frac{v-1}{2}$, then for each $d \neq 0$ there are exactly $\frac{k-1}{2}$ pairs $x, y \in D$ with $x - y = d \mod v$; thus there are exactly $\frac{k+1}{2}$ pairs $x \in D, y \notin D$ with $x - y = d \mod v$. So if b is a binary sequence of length v with $b_i = 1$ precisely when $i \in D$, we have

$$\mu(b) \geq \frac{k+1}{2} = \frac{v+1}{4} \tag{13}$$

We can have strict inequality in (13), because we may have $y = x - d < 0$ when the indices are not taken modulo v, giving a realization of d even though $y \in D$.

A *translate* of D is a set $D + t = \{a + t | a \in D\}$. Clearly any translate of a difference set is another difference set with the same parameters, but different translates can give different values of $\mu(b)$. However, we have $\mu(b) \leq 1 + \frac{v+1}{4}$ because $\mu_1 \leq 1 + \frac{v+1}{4}$; the only way to get a non-cyclic realization of $d = 1$ is at $b_0 = 1$ when $b_{v-1} = 1$. Experimentation suggests the following

Conjecture 1. for every difference set of size v, there exists a translate with $\mu(b) = 1 + \frac{v+1}{4}$.

Going in the other direction, a difference set always gives us a sequence b attaining $\mu(b) = q$ with length strictly less than $4q - 1$; taking a translate with $b_{v-t} = b_{v-t+1} = \cdots = b_{v-1} = 0$ lets us truncate b to length $v - t = 4q - 1 - t$.

Cyclic difference sets do not take advantage of the edge effects inherent in this problem, and they do not necessarily provide optimal solutions. It appears to be possible to do somewhat better than $\frac{v+1}{4}$ for all v (see section 3.3), although it also seems that the maximum $\mu(b)/v$ approaches $\frac{1}{4}$ as v approaches infinity.

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

Note that most of these optimal values cannot be attained by the difference-set constructions of section 3.1. For example, a difference set of size 31 would give $\mu(b) \leq 9$. Furthermore, a sequence attaining $\mu(b) = 10$ with $v = 31$ could not be obtained by truncating a block of consecutive zeros from a larger difference set; the larger difference set would have $\frac{v+1}{4} = 10$ thus $v = 39$, but no cyclic difference set of that size exists.

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 (except 21 and possibly 35) exhaustive search confirms that there are no optimal vectors with weight less than $\frac{v}{2}$.

v	$\max \mu(b)$	an optimal vector
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 1 0 0 1 1 0 0 0 1 0 1 1 0 0 1 0 1 0 1 1 1

Table 1. Optimal windows for small v

3.3 Iterative Improvement of Windows

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 $\mu(b^{i+1}) > \mu(b^i)$ 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 μ_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 $\mu_d = \mu(b^i)$. 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 μ_d decreases to $\mu(b^i) - 1$). 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 μ_d of flipping bit i. Table 2 gives the strengths of the best vectors found in this manner.

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

Table 2. Best known $\mu(b)$ for large v

4 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 μ_d; instead we seek to maximize the $(r' + 1)^{\text{th}}$ 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 $\mu_{(d_1,d_2,\cdots d_r)}$ as the number of indices j where $b_j = 1, b_{j+d_1} = \cdots = b_{j+d_r} = 0$, and maximize $q_r(b) = \min \mu_d$ 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. Trivialy we have

$$q_r(b) \leq \frac{v + r}{r + 1} \tag{14}$$

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 $q = q_1$.

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

Acknowledgment

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

References

1. Mihir Bellare, Ran Canetti, and Hugo Krawczyk. Message authentication using hash functions: the HMAC construction. *CryptoBytes*, 2(1):12–15, Spring 1996.
2. Mark Luk, Adrian Perrig, and Bram Whillock. Seven cardinal properties of sensor network broadcast authentication. In *SASN '06: Proceedings of the fourth ACM workshop on Security of ad hoc and sensor networks*, pages 147–156, New York, NY, USA, 2006. ACM.
3. Jr. Marshall Hall. *Combinatorial theory (2nd ed.)*. John Wiley & Sons, Inc., New York, NY, USA, 1998.
4. A. Perrig, R. Canetti, D. Tygar, and D. Song. The TESLA broadcast authentication protocol. *CryptoBytes*, 5(2):2–13, Summer/Fall 2002.
5. Adrian Perrig, Ran Canetti, J. D. Tygar, and Dawn Xiaodong Song. Efficient authentication and signing of multicast streams over lossy channels. In *IEEE Symposium on Security and Privacy*, pages 56–73, 2000.