Efficient Message Authentication for Spread Spectrum Wireless Communications

Dr. Charles Boncelet
boncelet@udel.edu

David Carman
David_Carman@nai.com
<table>
<thead>
<tr>
<th>1. REPORT DATE</th>
<th>01 DEC 2007</th>
<th>2. REPORT TYPE</th>
<th>N/A</th>
<th>3. DATES COVERED</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>4. TITLE AND SUBTITLE</td>
<td>Efficient Message Authentication for Spread Spectrum Wireless Communications</td>
<td>5a. CONTRACT NUMBER</td>
<td></td>
<td>5b. GRANT NUMBER</td>
<td></td>
</tr>
<tr>
<td>6. AUTHOR(S)</td>
<td></td>
<td>5c. PROGRAM ELEMENT NUMBER</td>
<td></td>
<td>5d. PROJECT NUMBER</td>
<td></td>
</tr>
<tr>
<td>7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)</td>
<td>University of Delaware</td>
<td>5e. TASK NUMBER</td>
<td></td>
<td>5f. WORK UNIT NUMBER</td>
<td></td>
</tr>
<tr>
<td>8. PERFORMING ORGANIZATION REPORT NUMBER</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)</td>
<td></td>
<td>10. SPONSOR/MONITOR’S ACRONYM(S)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11. SPONSOR/MONITOR’S REPORT NUMBER(S)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12. DISTRIBUTION/AVAILABILITY STATEMENT</td>
<td>Approved for public release, distribution unlimited.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13. SUPPLEMENTARY NOTES</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14. ABSTRACT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15. SUBJECT TERMS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16. SECURITY CLASSIFICATION OF:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>a. REPORT</td>
<td>unclassified</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>b. ABSTRACT</td>
<td>unclassified</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>c. THIS PAGE</td>
<td>unclassified</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>17. LIMITATION OF ABSTRACT</td>
<td>UU</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>18. NUMBER OF PAGES</td>
<td>45</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>19a. NAME OF RESPONSIBLE PERSON</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std Z39-18
Problem Studied

• *Message authentication* is used to confirm the integrity of a message and the authenticity of its sender.

• Conventionally, message authentication is an *application layer* problem.

• For the class of communications systems considered here, we present message authentication solutions that work at the *physical layer*. 
Why?

• Primary reason to consider the physical layer is to reduce communications costs:
  – Saves power
  – Fewer bits or chips needed
  – Better control over false acceptance versus false rejection tradeoff
Spread Spectrum Communication

• Widely used, especially in military communications
• Each bit is represented by multiple chips
• Sender and receiver use same chip sequence to construct bits
• Spreading gain is the number of chips per bit.
Example: Spread Spectrum

Input bits

Chips

Resultant Waveform
Conventional MAC's

• A *Message Authentication Code* (MAC) is a sequence of bits that depends on message and secret key.

• Since only sender and receiver know key, only sender and receiver can create correct MAC.

• In wired networks, MAC's are 128 or more bits long.

• In wireless networks, MAC's can be much shorter.
Minitag Idea

- Our idea, which we call a minitag, is to use a sequence of chips, not bits, to represent the MAC.
- The MAC will consist of many chips, but not all have to be received correctly.
- (Good thing, since chip error rate is much higher than bit error rate.)
Message Authentication at Sender

- Sender computes tag using secret key
Authentication at Receiver

- Receiver computes tag:

![Diagram showing the process of authentication at the receiver](image)
Compare Tags to Verify

• Receiver compares tags to verify message authentication:

```
\[\begin{array}{c}
\text{S-Tag} \\
\text{R-Tag} \\
\text{XOR} \\
\text{Sum} \\
\text{T} \\
\text{Reject} \\
\text{Accept}
\end{array}\]
```
Minitag Analysis

• If message is correct, then all chips agree (except for noise)
• If message is false, then chips disagree with probability 0.5
• Assume Gaussian noise with variance depending on SNR
• Hypothesis becomes: choose between $$(0.5)^n$$ and $$p^l(1-p)^m$$ where $$l = \#$$ errors, $$m = \#$$ correct, $$p =$$ chip error probability
Chip Error Rate

$P(\text{Chip Error})$ vs. BER
For various values of chips per bit

\begin{align*}
\text{BER (before ECC)} & \quad 1\times10^{-6} \quad 1\times10^{-5} \quad 1\times10^{-4} \quad 1\times10^{-3} \quad 1\times10^{-2} \\
P(\text{Chip Error}) & \quad 0.1 \quad 0.2 \quad 0.3 \quad 0.4 \quad 0.5
\end{align*}

- 1024
- 256
- 64
- 16

NATO Crosslayer Workshop – June 2-3, 2004
APPROVED FOR PUBLIC RELEASE
Mintag Length vs. P(False Accept)

Mini-tag Length vs. P(False Alarm)
BER from $1\times 10^{-8}$ to $1\times 10^{-3}$, Coding gain = 64
Example: IS-95 CDMA

• Assume BER = 0.001 (before ECC), coding gain = 64, rate 1/2 ECC
• For $1 \times 10^{-7}$ security, a conventional tag needs 24 bits
• Chips needed = $24 \times 2 \times 64 = 3072$
• Minitag needs 1195 chips, a savings of almost two-thirds
Soft Decision Minitag

- Can use soft decision decoding
- Treat each chip as a Gaussian RV
  - If message is correct, all means = -1
  - If message is false, means randomly alternate between -1 and 1
- Use central limit theorem for analysis
Log Likelihood Ratio

Densities Under H0 and H1

And Log Likelihood Ratio

Densities

- H1
- H0

Log Likelihood Ratio

- l(x)
Soft Decision Performance

• Continuing IS-95 example, chips needed reduced to about 774, a savings of almost a factor of 4 from the original.
Conclusions

• By considering authentication at physical layer, reduced communications cost for message authentication by about 2/3 to 3/4

• Furthermore, we can tune false acceptance and false rejection probabilities

• Future work improving and extending to other communication scenarios.
Thanks

- Research supported by Communications and Networks Consortium, U.S. Army Research Laboratory, agreement DAAD19-01-2-0011
- The views and conclusions contained in this document are those of the authors and should no be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U. S. Government
DSSS Work

• At last year's review, presented work using a “minitag” for authentication.
• The minitag used chips, not bits, by altering the spreading sequence.
• Eg. Assuming coding gain of 64, BER=1e-3, false alarm and miss probabilities = 1e-7, rate=1/2 ECC,
  – Conventional needs 3072 chips
  – Minitag needs 1195 chips
Current Work on Minitag

• Extending to soft decision.
• Ex., can reduce chip length to 772 chips, a factor of 4 reduction from original 3072 chips.
• (Problem is that our analysis uses the central limit theorem, which may be inaccurate at the very small probabilities needed here.)
New Work

• Message authentication is a one bit process:

   *Essentially, one decides whether or not the message is authentic.*

• Also, want to extend to other modulation schemes, not just DSSS.
Idea: Treat bits as a group

• Send $n$ bits as a group. Make a single decision on the group.
• Can do hard or soft decision of each bit in the group.
• Advantages: simplicity, better performance than other methods, applies to many modulation methods.
Traditional vs. Soft-Decision Message Authentication

### Traditional Message Authentication

- **Transmitter**
  - Apply Conventional MAC on Message
  - Apply FEC on Message + MAC

- **Receiver**
  - Perform ECC on Message + MAC
  - Verify Traditional MAC On Corrected Message

### Soft-Decision Message Authentication

- **Transmitter**
  - Apply Conventional MAC on Message
  - Apply FEC on Message only

- **Receiver**
  - Perform ECC on Message
  - Perform Soft-Decision MAC Verify On Corrected Message
Hard Bit Tag

• Normally one specifies $P(\text{False Accept})$, e.g., $2^{-s}$ for $s$ bits of “power”

• Then one minimizes $P(\text{False Reject})$.

• Instead of doing ECC, do the following:
  – Transmit $n$ tag bits
  – If $k$ or fewer errors, accept; else, reject.

• Example: 48 bits of power can be achieved with $(n,k) = (48,0), (54,1), (59,2), (64,3), (68,4)$, etc.
Hard Bit Tag Performance

![Graph showing the performance of hard bit tags with different parameters.

- Blue line: (32,0)
- Green line: (46,3)
- Red line: (61,7)
- Cyan line: (84,14)

The graph plots the probability of error rate (Pr(FR)) against the signal-to-noise ratio (Eb/No) in dB.]
Hard Tag Detailed Performance
Hard Tag vs. ECC
Hard Bit Tag (ctd.)

• Hard Bit Tag is *extremely* simple to implement: Generate $n$ bits and count the number of bits in error.

• The Hard Bit Tag will outperform *any* (hard decision) ECC based scheme of same length.
Soft Tag

• Instead of hard decoding each bit, do soft decoding.
• If message is correct, $X \sim N(-1, \sigma^2)$
• If message is incorrect,
  $X \sim 0.5N(-1, \sigma^2) + 0.5N(1, \sigma^2)$

• Log likelihood ratio is $l(X) = \log(e^{ax}+1)/a$
• Where $a = 2/\sigma^2$
Soft Tag Performance

• Use numerical techniques to compute density of \( l(X_i) \)
• Convolve to get density of sum of \( l(X_i) \)
• Difficult computation since desired error probabilities are very small, e.g., \( 2^{-32}=2e-10, \ 2^{-48}=3e-15 \)
Soft Tag Performance

![Graph showing Soft Tag Performance](image)
Message Authentication via Non-Spread Soft-Decision Decoding

- Original concept reduced tag size and increased tag reliability by performing hard-decision decoding of new spread spectrum-based waveforms
- Problem: Some communications systems may not be amenable to spreading or the burden of cross-layer packet decoding
- **Alternate goal**: Apply message authentication in such a way that is more generalizable
- **Concept**: *Perform soft-decision decoding of a traditional uncoded message authentication tag*
Soft-Decision Decoding Approach

• Traditional Approach:
  – Demodulate, soft/hard decode, hard correct message and tag bits
  – **Verify using hard-corrected message and tag bits**
    • Do computed and received tags match bit-for-bit?

• Soft-Decision Decoding Approach:
  – Demodulate, soft decode, hard correct message bits
  – **Verify using hard-corrected message and soft-decoded tag bits**
    • Do computed and received tag bit values match “close enough”?
Soft-Decision Verification Security

• Two ways soft-decision verification can incorrectly mark an incorrect message as authentic (false accept failure):
  – Failure 1. Incorrectly received message results in the receiver computing a hard-decision MAC tag that is a hard bit-for-bit match (“collision”) with the received authentication tag (same as traditional message authentication risk)
  – Failure 2. Incorrectly received message results in the receiver computing a hard-decision MAC tag that is not a hard bit-for-bit match, but using soft-decision verification is “close enough”

• Our Security Approach
  – Make sure probability of either of the two events is less than the desired probability of forgery
Addressing “Hard” Collision Security

• To guarantee resilience against traditional collisions (Failure 1), we propose to generate and verify an authentication tag that contains at least $n_{min}$ bits, where:

  Desired $Pr(False Accept) = 2^{-n_{min}}$

  **Example:**
  
  If the Desired $Pr(False Accept) = 2^{-48}$, then
  
  $n_{min} = 48$ bits

• Total tag size is $n = n_{min} + n'$
Addressing “Soft” Collision Security

- To guarantee resilience against soft-decision collisions (Failure 2), we propose to generate and verify an authentication tag that contains:
  \[ n = n_{\text{min}} + n' \text{ bits} \]
- Next we determine \( n' \)
- Our soft-decision must evaluate two hypotheses:
  \[ H_0 \text{ (authentic): } X_i \sim N(1, \sigma^2) \]
  \[ H_1 \text{ (not authentic): } X_i \sim 0.5*N(1, \sigma^2) + 0.5*N(-1, \sigma^2) \]
  - where \( X \) is an unbounded continuous value where 1 indicates that the received and computed bits match, and -1 indicates that they do not match
  - \( \sigma^2 \) is dependent on the signal-to-noise ratio
Sample Means for the Two Hypotheses

• As a practical matter, $X_i$ will be bounded by 1 to -1, so revise $H_0$ s.t.:
  \[ _0 = (1-.68*\sigma^2) \text{ (assuming BPSK)} \]

• Mean of $H_0$ for all $n$ samples is:
  \[ n_0 = n(1-.68*\sigma^2) \]

  and for $H_1$,
  \[ n_1 = 0 \]

• However, the worse case forgery condition is a hard-decision bit-by-bit collision, so to be conservative, set
  \[ n_1 = n_{\text{min}}(1-.68*\sigma^2) \]
Setting the Verification Threshold

• The simplistic approach is to set the threshold at the *midpoint* between the means of the two hypotheses, this way the false accept (verify bad message) and false reject (reject good message) rates of the verification function are the same.

• Thus, the threshold is:
  \[
  t = \_ * ( n_0 + n_1 )
  \]
  \[
  t = n_{min} + n' * (1 - .68*\sigma^2) / 2
  \]
Authentication Tag Size Determination

• Since we are assuming AWGN and normal distribution, the z value that corresponds to a probability of forgery of $2^{-48}$ is 7.79

• Thus,

$$n' \times (1 - 0.68 \sigma^2) / 2 = 7.79 \times \sigma^2$$

• Solving for $n'$:

$$n' = \frac{2 \times 7.79 \times \sigma^2}{(1 - 0.68 \times \sigma^2)}$$
Example of Traditional Method

• Assume we wish to authenticate a 16-bit message with probability of forgery per attempt of $2^{-48}$

• Traditional method:
  – Generate and append 48-bit MAC tag
  – Generate and append 63 parity bits using a Binary BCH block code with $n = 127$, $k = 64$, $t = 10$ errors
  – Communicate 127 bits
Example of Soft-Decision Authentication Method

• First, determine $n'$ by selecting the worst signal-to-noise ratio that we expect to verify messages

  – Since the (n=127,k=64,t=10) BCH code can correct up to 10 errors, assume our worst case probability of bit error:

    $$p_E = \frac{10.5}{127} = 0.083$$

  – For BPSK, $E_b/N_0 = -0.15$ dB

  – Thus, $n' = \text{ceiling}(12.45) = 13 \text{ bits}$
Soft-Decision Message Composition

• So for the same 16-bit message and probability of forgery = $2^{-48}$:
  – Reduced packet size approach:
    • Generate and append a $48+13 = 61$ bit MAC tag
    • Generate and append 15 bits using a $(n=31, k=16, t=3)$ Binary BCH code
    • Communicate **92 bits**
      • Less bits than traditional method with at least same security and modestly better reliability
  – Increased packet reliability approach:
    • Generate and append a $48+16 = 64$ bit MAC tag
    • Generate and append 47 bits using a $(n=63, k=16, t=11)$ Binary BCH code
    • Communicate **127 bits**
      • Same bits as traditional method with at least same security and much better packet reliability
Soft-Decision Authentication Plans

• Remainder of FY04
  – Analytically examine the soft-decision authentication approach for various
    • Bit/packet error rates
    • Packet/message sizes
    • Security levels
  – Simulate the soft-decision authentication approach for various
    • Bit/packet error rates
    • Packet/message sizes
    • Security levels
Output

- Submitted paper to Milcom (acceptance pending), paper to NATO workshop (accepted).
- In process of writing 1-3 journal articles.
- Developing software to analyze and simulate these tags.