Efficiency Tests Results and New Perspectives for Secure Telecommand Authentication in Space Missions: Case-Study of the European Space Agency

Franco Chiaraluce, Ennio Gambi, and Susanna Spinsante

We discuss some typical procedures to measure the efficiency of telecommand authentication systems in space missions. As a case-study, the Packet Telecommand Standard used by the European Space Agency is considered. It is shown that, although acceptable under well consolidated evaluation suites, the standard presents some flaws particularly in regard to the randomness level of the pre-signature. For this reason, some possible changes are proposed and evaluated that should be able to improve performance, even reducing the on-board elaboration time.

Keywords: Authentication, cryptography, security, space missions.

Manuscript received Aug. 19, 2004; revised Feb. 28, 2005.

Franco Chiaraluce (phone: +39 0712204467, email: f.chiaraluce@univpm.it), Ennio Gambi (email: e.gambi@univpm.it), Susanna Spinsante (email: s.spinsante@univpm.it) are with Dipartimento di Elettronica, Intelligenza Artificiale e Telecomunicazioni (DEIT), Università Politecnica delle Marche, Ancona, Italy.

I. Introduction

Among the actions that can cause a failure of a space mission, a prominent role is played by the transmission of illicit commands from unauthorized operators; as a consequence, telecommand (TC) authentication is a very important issue [1]. Illicit TCs can be inserted in unprotected links and take advantage of the lack of an appropriate authentication scheme in order to disturb the mission. Accidental contributions, like noise or human errors, are often combined with intentional interference such as jamming. Moreover, the receiver of TC systems employing classic modulations (carrier phase modulation with digitally modulated sub-carriers) can lock onto unintentional interferences as well.

In this paper, we are primarily concerned with the possibility that malicious actions could be performed against space systems. External attacks can aim to violate confidentiality in the transmitted data (passive traffic analysis), prevent the providing of a service (as with RF interferences), and/or insert illegal TCs in order to modify or reply with intercepted legal ones (impersonation attacks). Actually, attacks of the third kind are the most dangerous and require special care in TC authentication. The problem seems to be becoming more and more important since current links are prone to extensively use ground stations interfaced with open networks (for example, the Internet) intrinsically characterized by high vulnerability.

Following the procedure also used in other commercial applications, authentication consists of adding a digital signature to the transmitted data. Such a signature is generated at the transmitting side by subjecting any message to non-linear

transformations able to map a variable number of bits in a standard and fixed length sequence. This way, each signature is univocally related to the content and sequence of the data, being at the same time highly unpredictable because of the random nature of the mapping. A long secret key is used for this purpose; the same secret key is also available on board the satellite. The signature is therefore computed again, and the received TC is accepted if and only if the calculated signature coincides with the received one. Otherwise, the message is blocked from further distribution.

An extensive body of literature exists on the security of authentication algorithms [2], [3]. For better evidence, however, in the remainder of the paper we will refer to the authentication system adopted by the European Space Agency (ESA) [4], [5]. The ESA algorithm is also referenced in the Consultative Committee for Space Data Systems (CCSDS) Green Book on general security options [6], so we can say it is a useful benchmark for the international community. The ESA authentication system will be shortly reviewed in section II. In section III, we will present some typical tools for measuring the efficiency of an authentication procedure, and in section IV we will apply these tools to the ESA algorithm. The results will show that, while generally effective, the ESA Packet Telecommand Standard has some margins for improvement, either in terms of a fairer distribution of the randomization action (which is the key for success in the authentication algorithm) among the blocks of the signature generator or in the ability to face some critical situations (like the transmission of very short TCs). Moreover, it seems advisable to design authentication algorithms that require shorter and shorter processing times in such a way as to maintain as high as possible the throughput of the link. For these reasons, in section V we will propose some possible modifications to the ESA standard that, while maintaining basically unchanged the structure of the authentication system, should permit us to improve its efficiency. An explicit evaluation of the advantage achievable will be provided in section VI. Finally, section VII will conclude the paper.

II. The ESA Authentication System

A schematic representation of the signature generator used in the ESA Packet Telecommand Standard is shown in Fig. 1. For the sake of brevity, in this section the main functionalities of this scheme are shortly described; details (necessary for practical implementation) can be found for instance in [5].

In Fig. 1, m is related to the TC frame data field; it is obtained through the concatenation of the TC, the contents and identifier (ID) of a logical authentication channel (LAC) counter, and a variable number of stuffing zeros; $S = f(m)$ is the corresponding signature.

The hash function, used to reduce the data field into a 60-bit value called a pre-signature (P), is realized via a linear feedback shift register (LFSR) that is 60 bits long, whose feedback coefficients are programmable but secret as they are part of the authentication key. This way, the pre-signature is kept secret as well. The LFSR is shown in Fig. 2: sums are modulo 2 (XOR), while the feedback coefficients cf_i are taken from the secret key.

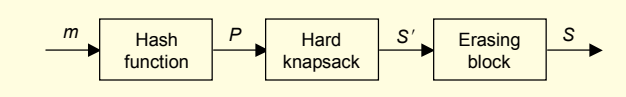

Fig. 1. Main functions of the ESA signature generator.

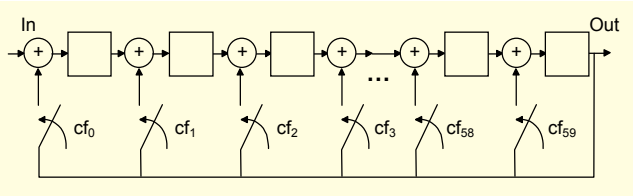

Fig. 2. LFSR implementation of the hash function.

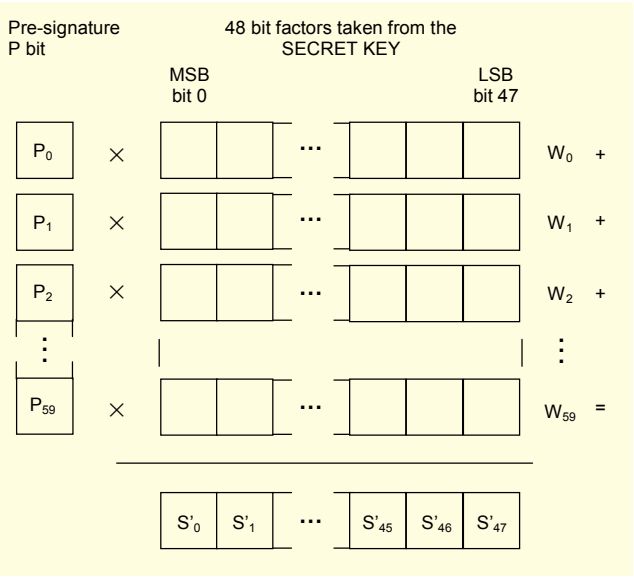

Fig. 3. Implementation of the hard knapsack function.

The hard knapsack is a transformation process which involves the multiplication of each incoming bit with one out of 60 different vectors, 48 bits long, which are secret in their turn, and the final sum of the results. The operation is shown in Fig. 3; the result is a preliminary version of the signature, noted by S' and computed as follows:

$$S' = \left(\sum_{j=0}^{59} P_j \cdot W_j \right) \bmod Q \,, \qquad (1)$$

where $Q = 2^{48}$, W_j is the j-th hard knapsack factor, and P_j the j-th

bit of the pre-signature. Signature *S'* is truncated in such a way as to have a final signature *S* that, unlike *S'*, is 40 bits long. This is done by the erasing block, which eliminates the 8 least significant bits of the knapsack output.

It should be noted that the secret key for authentication consists of $60 + 60 \times 48 = 2940$ bits, which is a very large value, particularly if we consider that the key can be periodically updated. This process is essentially "one - way", as most of the input data is lost during transformation; the same input will always result in the same signature, but a great number of different inputs can yield an identical output.

III. Evaluation Tests for Authentication Systems

For an authentication algorithm to be efficient and robust, function $S = f(m)$ must satisfy at its best a number of properties [7]; among them,

a) it should be easy to compute *S* from the knowledge of *m*;
b) for a given *m*, it should be difficult to find another message $m' \neq m: f(m') = f(m)$;
c) it should be difficult to compute the secret key from the knowledge of *m* and *S*; and
d) it should be difficult to generate a valid *S* from the knowledge of *m*, without knowing the secret key.

In this list, "easiness" and "difficulty" are related with the time required for executing the algorithm; this in turn grows with the data dimension. So, we can say that an algorithm is *computationally easy* when the processing time it requires grows according with a polynomial (for example, linear) law; on the contrary, an algorithm is *computationally difficult* when the processing time grows according with a much more severe (for example, exponential) law. In the latter case, the processing time and the other resources needed (for example, memory) become rapidly unfeasible. From a different point of view, this introduces the problem of "complexity", and in this sense it is worth mentioning that the ESA authentication algorithm to which we refer was formulated more than 15 years ago, when much fewer computational resources were available. Therefore, a complexity evaluation must take into account the progress that has occurred for computers in the meantime.

Coming back to the list of features above, property a) is ensured by the structure in Fig. 1, which is very simple to implement. Property b), rather, is generally satisfied if the signature exhibits the typical features of a random sequence (that is, uncorrelation and equiprobability). Properties c) and d) are also safeguarded by a good choice of the secret key.

The randomness level reached can be measured, for example, through some of the tests included in a well-known suite by the National Institute of Standards and Technology (NIST) [8].

Statistical tests, such as the NIST test, give a first indication of security. In addition, other tests more specific for verification of the functions adopted in an authentication algorithm are available [9]; in particular,

- the bit variance test measures the probabilities of taking on the values 1 or 0 for each bit of the signature produced;
- the entropy test measures the entropy *H* of the signature generator; and
- the cross-correlation test measures the cross-correlation function *M* between different signatures.

The bit variance test and the cross-correlation test focus on the evaluation of non-linearity properties of the result; the random behavior of the signature allows the opponent to perform only a brute-force attack on the generator; while on the other hand, the entropy test measures the probability that a particular signature is generated, thus evaluating the complexity level of the generator, which is the main issue against attacks based on the so-called "birthday paradox" [10]. These tests have been originally specified with reference to the hash function, but they can be obviously extended to the other parts of the signature generator. The term "digest" is usually used synonymous of signature in this more general sense. The bit variance test measures the uniformity of each bit of the digest. For this purpose, once an input message is chosen, its bits are varied according with the procedure described below, and for each bit of the digests correspondingly produced, the probabilities $\Pr(\cdot)$ of taking on the values 1 and 0 are evaluated. The bit variance test is passed if $\Pr(S_i = 1) \approx \Pr(S_i = 0)$ for all the bits S_i of the signature, with $i = 1 \ldots n$, where n is the signature length. The input message bit-changes can be obtained by EX-ORing ($\oplus$) the original input message with Boolean vectors α of length n and Hamming weight $W_H(\alpha)$, ranging in principle from 1 to n. Actually, as it is computationally very hard to consider all the input message bit-changes, a maximum value of $W_H(\alpha)$, denoted by k, is normally selected, and the test is applied in a reduced but practical scenario. In conclusion, the generating function passes the bit variance test if it satisfies the following "propagation criterion of degree k":

$$\begin{aligned} &\forall \alpha \in F_2^n : 1 \le W_H(\alpha) \le k, \quad k \le n \\ &\Pr\left(g_i(X) = g_i(X \oplus \alpha)\right) \approx \frac{1}{2}, \end{aligned} \tag{2}$$

where F_2^n is the set of all Boolean vectors with length n, $i = 1,\ldots,n$, and $g_i(X)$ represents the i-th bit of the digest obtained by hashing the input message X.

With the entropy assessment, the uniformity of each block of bits in the digest is tested. Entropy measures the information content of a sequence, which is at maximum when it equals the number of bits constituting it; in this situation, all sequences are uncorrelated and have the same probability to occur. By applying these classic notions to the signatures generated by the functions in Fig. 1, the maximum entropy value obtainable is 40. In such a specific context, entropy should be calculated by computing the probability of occurrence of any possible 40-bit signature, S^i, over the set of signatures, with cardinality 2^{40} produced by feeding the hash function with different input messages. Formally,

$$H\left(S^1, S^2, ..., S^{2^n}\right) = -\sum_i \Pr(S^i)\log_2\left[\Pr(S^i)\right] \quad \forall i : 1 \le i \le 2^n \ . \tag{3}$$

To make the test significant, a very large number of messages, and then signatures, should be considered. Then, similarly to the bit variance test, the computational effort required for computing (3) could be intolerably high. The birthday paradox approach allows the reduction of this effort, but it does not make the calculation feasible yet. In order to evaluate an approximate entropy, we have adopted the alternative method suggested in [9]. The digest, *n*-bits long, is decomposed into *t* blocks of *r* bits each (*t* and *r* suitably chosen). Each block is represented by an integer ranging between 0 and $2^r - 1$. The entropy of the blocks is then computed, which is equal to *r* if they are equally probable. And the entropy test is considered (approximately) passed in this case. Such a procedure has the advantage to use each digest generation to produce *t* contributions for the calculus; moreover, in [9] it was stated that it has almost the same validity of the complete test.

Finally, the cross-correlation test measures the cross-correlation function *M* between different signatures. Given two bipolar sequences f^1 and f^2 with the same length *n*, the cross-correlation *M* is defined as

$$M = \frac{1}{n}\sum_{i=1}^{n} f_i^1 \cdot f_i^2 \ . \tag{4}$$

According with this definition, the cross-correlation of the two sequences is in the range [−1, 1]; when the sequences differ for all the *n* bits, *M* is −1, while it takes the value 1 in the opposite case of equal sequences. Two sequences which differ for half the bits have $M = 0$. In order to use this parameter as a further measure of the efficiency of an authentication algorithm, it seems meaningful to compute the cross-correlation values of signatures resulting from strongly correlated input messages. If the authentication algorithm performs well, similar messages should generate strongly uncorrelated signatures; the ideal condition for them is therefore $M << 1$.

Besides the one previously described, other techniques could be used for security evaluation, based for example on "strong" attacks, such as linear and differential attacks or internal collision attacks. However, interpretation of the results of these further tests for authentication algorithms is, at our knowledge, not yet fully clear, and the procedures we have applied seem quite satisfactory for the first order analysis here proposed.

At the end of this section, it should be noted that we have deliberately decided not to address here the effect of the secret key choice. Discussion on the secret key is postponed to section V, where the choice adopted in all our tests, based on a minimum cross-correlation rule, will be presented in detail. Indeed, investigation of different rules is an important research topic we are currently working on.

IV. Performance Evaluation of the ESA Authentication System

According with [5], in simulating the ESA authentication system, the hash function has been implemented as an LFSR with 60 feedback coefficients selected from the last 60 bits of the authentication key; the initial configuration (seed) of the LFSR is specified in [5]. The efficiency of the adopted authentication procedure is appreciated better if the features of the intermediate results are determined, together with those of the final signature; therefore, tests have been applied to the pre-signature *P* (60 bits), the non-truncated signature *S'* (48 bits), and the final signature *S* (40 bits) generated by the authentication algorithm. This way, one is able to appreciate the balance of the randomization action (which is the result of a number of operations) among the various parts of the generator.

Although the number of possible configurations to test is huge, in principle, in the following we will discuss the performance of the authentication system in an operation condition characterized by the repeated transmission of the same (generic) TC. This is because such a situation can be considered as the most critical one from the security viewpoint, giving to an opponent a large number of plaintext/signature associations to test when he tries to recover the authentication key adopted. Under this condition, the authentication key and the TC have been fixed, while the LAC counter value has been changed. Starting from an initial value equal to 0, 25,000 signatures have been computed for the same TC value, simulating the repeated transmission of the same TC by increasing the LAC value of one unit at each step. We tried to give an evaluation of the system performance as realistically as possible by adopting different TC lengths (2, 4, 8, 32, 128, 184,

239 bytes) and by setting binary sequences with bit pattern repetitions as TCs. The length values tested agree with the ESA PSS standard presently applied, though the TC frame is likely to increase in the future by adopting updated CCSDS recommendations (to appear in [11] at a later date). Following the same strategy mentioned above, in order to test the system under the worst conditions, TCs with different lengths have been simulated privileging the choice to have similar patterns; for example, the maximum-length TC has been obtained by repeating, as many times as necessary, the same binary pattern of a shorter TC. The ESA authentication system can be considered good if the high correlation existing among the transmitted data frames is lost over the corresponding signatures.

First of all, the NIST Test Suite [8] has been applied. Each statistical test in the Suite, with the exception of the self-correlation test, returns a *p_value* that represents the probability that a sequence produced by a random number generator is less random than the sequence under test, according to a significance threshold *th*. If $p\_value \geq th$, then the sequence under test is assumed to be random with a confidence level of $[(1 - th)\cdot 100]\%$. Further details on the management of the NIST Test Suite and the significance of the various proofs can be found in [8], but are here omitted for the sake of brevity.

In Table 1, the results of some of the NIST tests over the signature *S* are reported, considering the seven different TC lengths (listed above) and a threshold $th = 0.01$. According with the criterion previously mentioned, the test occurred with success (Y) when the returned *p_value* was greater than 0.01; otherwise, the test did not have success (N). The corresponding confidence level is 99%. By looking at the table, it is possible to say that the statistical properties of the generated signatures get better as the TC length increases. All tests have been passed for lengths $\geq$ 128, while the same is not true for the shorter lengths. The longer sequences are more significant in testing the performance of the authentication algorithm since, because of the strategy adopted (repetitions of the same binary sequence), they are characterized by increasing correlation. So, we can conclude that the random properties of the produced signature are generally better for more correlated inputs, and the behavior of the ESA signature generator is globally good. Anyway, two specific points remain: a) when short TC lengths are adopted, some of the tests are not satisfied and b) if the same tests are applied not only to the final value generated by the authentication algorithm but also to the values produced by the intermediate steps of the authentication process, such as the pre-signature *P* or the non-truncated signature *S'*, they provide much worse results. In particular, we have not reported here any result about the pre-signature *P*, as it fails the first fundamental test (Frequency), thus making inapplicable all the others.

Besides the NIST Test Suite, the specific tests for authentication algorithms, described in section III, have also been applied. Precisely, the bit variance test has been performed on *P* and *S*, by considering the seven different TC lengths and up to five different Hamming weight values; this means that all possible messages differing from the generated *m* (related, in its turn, to a specific TC) by 1 to 5 bits have been considered. In Table 2, the mean probabilities of taking on the value 1 or 0, for the bits of the pre-signature and the signature, respectively, are shown. It is evident that, in all situations examined, the signature *S* shows a better behavior than that of

Table 1. Some of the NIST Test Suite results on the signatures S generated by the ESA authentication system, considering seven different TC lengths (result: Y = success, N = failure).

	TC2		TC4		TC8		TC32		TC128		TC184		TC239	
TEST	*p_value*	result	*p_value*	result	*p_value*	result	*p_value*	result	*p_value*	result	*p_value*	result	*p_value*	result
Frequency	0.563939	Y	0.680339	Y	0.909238	Y	0.698756	Y	0.678874	Y	0.678874	Y	0.855583	Y
Block frequency	0.115958	Y	0.677529	Y	0.544281	Y	0.338369	Y	0.555614	Y	0.555614	Y	0.465554	Y
Runs	0.55866	Y	0.721904	Y	0.626254	Y	0.984103	Y	0.745873	Y	0.745873	Y	0.96808	Y
Longest run of ones	0.311792	Y	0.22178	Y	0.349953	Y	0.071101	Y	0.398622	Y	0.398622	Y	0.934168	Y
Rank	0.218916	Y	0.74586	Y	0.603452	Y	0.283006	Y	0.47608	Y	0.47608	Y	0.655687	Y
Discrete Fourier transform	0.000411	N	0.001546	N	0.001808	N	0.001808	N	0.043502	Y	0.043502	Y	0.043502	Y
Random excursion	0.004084	N	0.814968	Y	0.958498	Y	0.25182	Y	0.346303	Y	0.346303	Y	0.336644	Y

the corresponding pre-signature P. This is reasonable and expected, since the final target of the generator device is to produce a good final signature. Anyway, if we look at the entire procedure as the result of a joint action, we could aim to also improve the features of the intermediate results. It is easy to understand, in fact, that this can contribute to make harder the task of an opponent trying to reconstruct the signature generation chain. In this sense, margins for improvement seem to exist in particular on the hash function adopted, as in certain cases the mean probability value for P is far from the desired one of 50.00%. Obviously, the cross-correlation test and the entropy test have been also applied to the ESA system providing, however, good results: in particular, the cross-correlation values for S are nearly zero, and the entropy test is largely satisfied as well.

Based on the results of this paragraph, we can say that the ESA authentication system is robust regarding the cross-correlation and entropy properties of the produced signature, and generally acceptable also from the NIST Test Suite viewpoint since it produces quasi-random sequences.

Some improvements, however, might be possible, taking into account the following remarks:

- The pre-signatures generated by the hash function may be strongly correlated to the initial data to be authenticated and do not have a suitable randomness level, as they fail the randomness tests of the NIST Suite. Under this condition, the goal of making the authentication scheme robust is mainly left to the hard knapsack block. It should be preferable to enforce also the other authentication system building blocks in order to ensure better performance under the security point of view.
- In the case of short TCs, even the results on the final signature S are questionable, and do not satisfy completely the NIST Test Suite. Because of the previous remark, such a conclusion must be seen as a criticism to the hard knapsack rule adopted, which becomes more and more vulnerable when increasing the quasi-periodic structure of the input.

Based on these observations, in the next section we present some simple changes to the hash function and the hard knapsack function adopted by ESA, which permit an improved performance and overcome as much as possible the evidenced drawbacks.

V. Proposals for a Modification of the ESA Authentication Scheme

Customized hash functions are those which are specifically designed for the explicit purpose of hashing, without being constrained to reuse existing system components such as block ciphers or modular arithmetic. RIPEMD128 [12] is a hash function based on Message Digest 4 (MD4), developed under the European Race Integrity Primitives Evaluation (RIPE) program. It differs from MD4 in the number of rounds (4 rather than 3), the parallel execution of two distinct versions of the compression function (the left and the right lines), the order in which the input words are accessed, and the amounts by which the results are rotated. RIPEMD128 takes a variable length input to produce a fixed length output of 128 bits.

The first operation performed is bit padding, according to the so-called Merkle-Damgård strengthening technique, so that the input length is congruent to 448 modulo 512. Next, the message is processed block by block by the underlying compression function, as schematically shown in Fig. 4. At each step, the following operation is realized: $A := (A+f(B,C,D)+X+K)^{<<s}$, where f is a logic function and $^{<<s}$ denotes a cyclic shift (rotation) over s positions. As the figure shows, four different logical functions are used in this solution,

Table 2. Mean probabilities (expressed in percent), from the bit variance test, using the ESA authentication system for different TC lengths.

	$W_H(\alpha)=1$		$W_H(\alpha)=2$		$W_H(\alpha)=3$		$W_H(\alpha)=4$		$W_H(\alpha)=5$	
	P	S	P	S	P	S	P	S	P	S
TC2	45.36%	49.60%	45.69%	49.95%	52.53%	50.01%	50.02%	50.00%	49.9%	49.99%
TC4	43.59%	50.19%	43.86%	50.08%	50.69%	50.01%	49.57%	49.99%	50.01%	49.99%
TC8	46.56%	49.73%	46.47%	49.85%	50.23%	49.98%	49.99%	50.00%	50.00%	50.00%
TC32	44.96%	50.06%	52.30%	50.00%	49.99%	49.99%	49.99%	50.00%		
TC128	45.90%	50.26%	49.76%	50.01%						
TC184	46.20%	50.20%	49.98%	50.00%						
TC239	45.98%	50.01%	50.04%	49.99%						

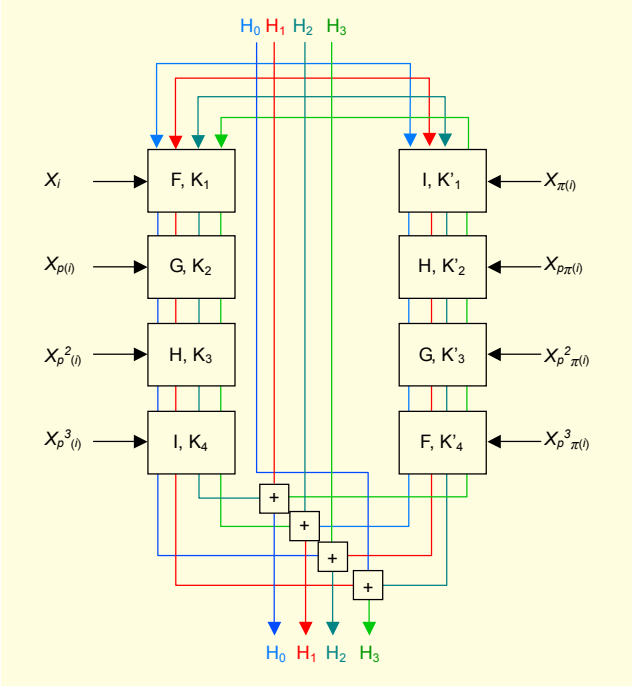

Fig. 4. Outline of the compression function of RIPEMD128.

Table 3. Primitive logical functions in RIPEMD128.

Step index	Function name	Function value
$0 \le j \le 15$	$F=f(j,B,C,D)$	$B \oplus C \oplus D$
$16 \le j \le 31$	$G=f(j,B,C,D)$	$(B \wedge C) \vee (\neg B \wedge D)$
$32 \le j \le 47$	$H=f(j,B,C,D)$	$(B \vee \neg C) \oplus D$
$48 \le j \le 63$	$I=f(j,B,C,D)$	$(B \wedge D) \vee (C \wedge \neg D)$

$\neg$: NOT, $\wedge$: AND, $\vee$: OR, $\oplus$: EX-OR

Table 4. Additive constants in RIPEMD128.

Step index	Left line	Right line
$0 \le j \le 15$	$K_1=K(j)$=00000000	$K'_1=K'(j)$=50A28BE6
$16 \le j \le 31$	$K_2=K(j)$=5A827999	$K'_2=K'(j)$=5C4DD124
$32 \le j \le 47$	$K_3=K(j)$=6ED9EBA1	$K'_3=K'(j)$=6D703EF3
$48 \le j \le 63$	$K_4=K(j)$=8F1BBCDC	$K'_4=K'(j)$=00000000

denoted by F, G, H, I and reported in Table 3. A 128-bit buffer is used to hold intermediate and final results of the hash function. It consists of four 32-bits registers (A, B, C, D) initialized to the following hexadecimal values: $A = H_0 =$ 67452301, $B = H_1 =$ EFCDAB89, $C = H_2 =$ 98BADCFE, and $D = H_3 =$ 10325476. Concatenation of these values provides the initial chaining variable CV_0 to be used either from the left or from the right line.

Updating of the chaining variable CV_i (with i describing the operation progress) takes place through independent processing by the two parallel lines, each one consisting of four 16-step rounds. Each round takes as input the current 512-bit block being processed (X_i and its versions suitably changed through permutations ρ and π) and the 128-bit buffer value $ABCD$ (left line), or $A'B'C'D'$ (right line), and updates the content of the buffer. Besides the reverse order in realizing the logic functions, the two processing lines also differ in the value of the additive constants, which are given in Table 4.

The output of the fourth round is added to the chaining variable input to the first round (CV_i) to produce CV_{i+1}. The addition is done independently for any of the four words in the buffer of each line, with the corresponding word in CV_i, by using a modulo 2^{32} sum, and involves rotation of the words of each of the three inputs. After all the 512-bit blocks have been processed, the output from the last stage is the 128-bit message digest. A more complete description (and further numerical details for implementation) can be found in [12].

The RIPEMD128 hash function, adopted in the framework of a possible new version of the ESA authentication system, generates a 128-bit pre-signature P, but only the first 60 MSBs (most significant bits) are considered, in such a way as to leave unchanged the length of P.

Another possible change to the ESA authentication scheme concerns the hard knapsack rule adopted; a proposal is reported below, together with a selection criterion for the multiplicative factors. A crucial point in establishing the features of the hard knapsack function concerns the choice of the multiplicative factors W_j. The sequence of the W_j's constitutes the most relevant part of the authentication key. In order to improve performance, these 48-bit coefficients can be selected on the basis of a minimum cross-correlation rule: only those factors satisfying well-precise conditions on the cross-correlation with the previous factors are accepted and inserted in the authentication key. In our implementation, a 60-bit LFSR, with maximum length feedback coefficients, has been used to generate the "candidate" multiplicative factors. The authentication key has been obtained by selecting the factors with cross-correlation in the range of [–0.167, 0.167], which means that any pair of 48-bit factors differ by 16 to 32 bits. About 6,200,000 factors have been generated before finding this "optimal" key. The computational effort would be lower in the case of a wider cross-correlation values range being accepted.

The best utilization of these optimized hard knapsack factors implies modification of the mathematical relationship the function is based on. Instead of (1), it is possible to use the following expression:

$$S' = \sum_{j=0}^{59} P_j \overline{\oplus W_j} \ , \qquad (5)$$

with the same meaning of the parameters involved. In practice, the inner product is now replaced with an EX-NOR logic function. This means that any (selected) factor W_j now gives a contribution to the signature S' (unchanged when $P_j = 0$ or complemented when $P_j = 1$), while in using (1) there was no contribution from W_j when $P_j = 0$. Because of the introduction of the logic function, the mod Q operation in (1) is no longer necessary.

The improved efficiency of the new hard knapsack rule can be soon verified by comparing the results of the bit variance test, in a simplified version of the authentication system where only the hard knapsack rule is modified according to (5). As shown in Table 5, the positive impact of the new rule is in terms of reduced dispersion of the values around the mean, particularly for the smaller Hamming weights. A more complete comparison, involving also the other tests and including the modified hash function, is presented in the next section.

VI. Performance Evaluation of the Modified Authentication Scheme and Comparison with the Original ESA System

An authentication system including the changes in the hash function and the hard knapsack function, as presented in section V, is plotted in Fig. 5.

As for the original ESA authentication scheme, performance evaluation of the modified system has been obtained by means of the NIST Test Suite in order to measure the randomness level of the generated signatures, and by means of the more specific statistical tests for authentication procedures.

In Table 6, the results of some of the NIST tests over the signature S are reported, considering different TC length values. By comparing these results to the ones obtained by testing the original ESA system, shown in Table 1, we can say that the modified version gives better performance, and even the tests that failed before are now successfully passed.

In Table 7, the results for the bit variance test using the modified new scheme and different TC lengths are also reported. Comparison with Table 2 shows a great improvement in the values relative to the pre-signature P, which are now everywhere closer to the ideal value of 50.00%; this is clearly a demonstration that the RIPEMD128 hash function gives better performance than the previous simple LFSR scheme. As a further result, we have also proved that, contrary to the previous situation (that is, the ESA authentication system), the pre-signature P passes the "Frequency" (and actually all of the) NIST Suite tests.

From the same comparison between Table 2 and Table 7, we also see that the behavior of S, in regard to the bit variance test, is sometimes worsened by the modification, showing a slightly larger dispersion around the optimal value.

This degradation, however, does not seem particularly worrying. It may be due to some uncertainty in the management of the statistical results (which is more critical when the system is close to 50%) and, even if confirmed, it is largely compensated by the improvement in the behavior of P. In other words, it is confirmed that substantial improvements can be obtained on the pre-signature, while preserving the good behavior of the signature, with very small modifications of the current standard.

Apart from the considerations on the advantage of a more balanced distribution of the randomization effort, already done in the previous sections, an improved behavior of the pre-signature P also yields an indirect benefit to the hard knapsack function. One of the main concerns in using this function, in fact, which also reflects on the security issues, is the basic linearity of its structure; a classic method to counterbalance linearity consists in increasing the randomness level of the input. This in particular permits us to obtain better performance under the "confusion" and "diffusion" points of view, which are two fundamental cryptological properties [13]. The improved hash function has therefore positive effects in this sense, too.

Table 5. Bit variance test comparison between the original and the new hard knapsack rule (TC length = 1 byte).

	$W_H(\alpha) = 1$		$W_H(\alpha) = 2$		$W_H(\alpha) = 3$		$W_H(\alpha) = 4$	
	Original hard knapsack	Modified hard knapsack	Original hard knapsack	Modified hard knapsack	Original hard knapsack	Modified hard knapsack	Original hard knapsack	Modified hard knapsack
Mean	49.14%	50.59%	49.87%	50.01%	49.99%	49.98%	50.00%	50.00%
Max	67.18%	62.50%	52.28%	52.43%	50.64%	50.29%	50.31%	50.14%
Min	29.69%	40.63%	46.32%	47.72%	49.60%	49.60%	49.81%	49.88%

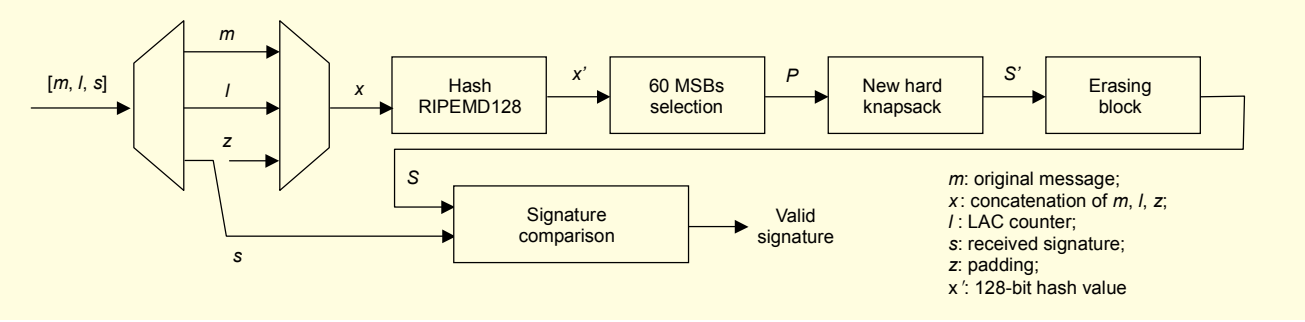

Fig. 5. Modified version of the authentication system.

Table 6. Some of the NIST Test Suite results on the signatures S generated by the modified authentication system, considering seven TC lengths (result: Y = success, N = failure).

	TC2		TC4		TC8		TC32		TC128		TC184		TC239	
TEST	*p_value*	result	*p_value*	result	*p_value*	result	*p_value*	result	*p_value*	result	*p_value*	result	*p_value*	result
Frequency	0.278808	Y	0.028452	Y	0.440707	Y	0.613559	Y	0.695058	Y	0.695058	Y	0.644799	Y
Block frequency	0.133817	Y	0.773777	Y	0.649844	Y	0.193991	Y	0.754532	Y	0.754532	Y	0.968226	Y
Runs	0.924781	Y	0.969369	Y	0.817546	Y	0.072834	Y	0.801755	Y	0.801755	Y	0.053842	Y
Longest run of ones	0.144048	Y	0.863114	Y	0.703841	Y	0.615212	Y	0.823599	Y	0.823599	Y	0.720152	Y
Rank	0.562908	Y	0.495526	Y	0.875915	Y	0.969425	Y	0.09686	Y	0.09686	Y	0.277623	Y
Discrete Fourier transform	0.926884	Y	0.613759	Y	0.926884	Y	0.520637	Y	0.581909	Y	0.581909	Y	0.926884	Y
Random excursion	0.021803	Y	0.210042	Y	0.32342	Y	0.73246	Y	0.603552	Y	0.603552	Y	0.689233	Y

Table 7. Results for the bit variance test using the modified authentication system and different TC lengths (mean values).

	$W_H(\alpha) = 1$		$W_H(\alpha) = 2$		$W_H(\alpha) = 3$		$W_H(\alpha) = 4$		$W_H(\alpha) = 5$	
	P	*S*	*P*	*S*	*P*	*S*	*P*	*S*	*P*	*S*
TC2	49.51%	50.98%	49.81%	49.72%	50.00%	49.78%	50.02%	49.97%	50.00%	50.00%
TC4	49.21%	50.70%	49.93%	49.80%	49.96%	49.98%	50.00%	50.01%	49.99%	50.00%
TC8	50.69%	50.33%	49.98%	49.92%	50.00%	49.97%	50.00%	50.00%	50.00%	49.99%
TC32	50.85%	49.53%	49.98%	50.02%	50.00%	50.00%	49.99%	50.00%		
TC128	49.65%	50.27%	49.99%	49.98%						
TC184	49.75%	49.91%	49.99%	49.99%						
TC239	49.89%	50.07%	50.00%	49.98%						

In Table 8, a comparison among the entropy values for the pre-signatures *P* obtained for different TC lengths, by means of the original and the modified authentication systems, is shown.

Given the software implementation of the entropy test used to evaluate the system performance, the test provides the best results if the entropy value obtained equals the block test length.

Since a block test length of 12 bits has been assumed in the simulation, the ideal entropy value is 12, which is difficult to be reached by the original system. On the contrary, it is practically achieved by the modified version.

In conclusion, the modified scheme seems to ensure full satisfaction of the tests for the pre-signature P and, in all conditions, for the final signature S (while some "critical" situations existed for the original scheme).

Finally, a further merit of the modified signature generator is in the reduced processing time required. Maintaining a limited processing time is particularly important in satellite applications. Although an objective measure of the processing time is unpractical, because of the great number of variable conditions that influence its evaluation, we can consider a relative estimation by measuring the time required for authentication via the modified algorithm (T_m) normalized to the same parameter for the original ESA algorithm (T_0). While the original algorithm is faster in authenticating short TCs, the situation is reversed in the case of longer TCs.

A plot of the normalized processing time is reported in Fig. 6.

Table 8. Entropy values of the pre-signatures P obtained for different TC lengths by means of the original and the modified authentication systems.

	Entropy	
	Original system	Modified system
TC2	7.72687	11.9996
TC4	8.38089	11.9997
TC8	8.38516	11.9996
TC32	8.41893	11.9996
TC128	8.74006	11.9996
TC184	8.67674	11.9996
TC239	8.77583	11.9996

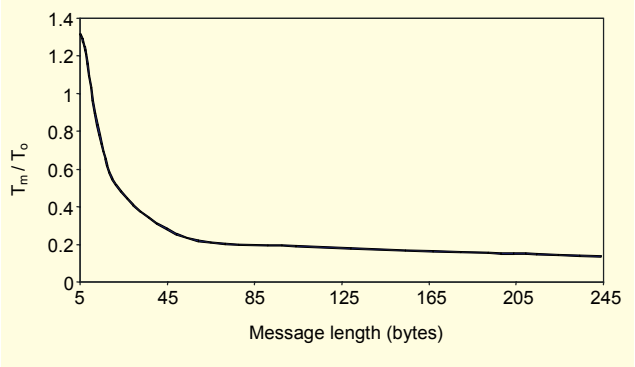

Fig. 6. Normalized processing time for the modified algorithm as a function of the message length.

The modified algorithm shows a processing time reduction of more than 80%, when the data frame lengths approach the maximum admitted values. This behavior can be explained by considering the difference in the implemented hash functions. The hash function performed via LFSR, in the original version of the authentication system, needs fewer operations on the data, but it acts on single bits, thus requiring longer processing times when the data frame length gets higher.

The RIPEMD128 function is more complicated in terms of processing operations on the data, but it can operate on 512-bit blocks, giving better performances as the length of data frames increases. Tests have also shown that most of the execution time is consumed by the hash function, while the hard knapsack implementation, both original and modified, is not so heavily time-consuming. Obviously, the suggested selection of the hard knapsack factors, on a minimum cross-correlation basis, is to be done off-line. This is also because the same factors, taken all together, determine the authentication key adopted, which must be established before all the authentication functions take place.

VII. Conclusions

In this paper we have presented a quantitative framework for measuring the security level of an authentication system for space mission applications. As a useful benchmark, we have considered the Packet Telecommand Standard adopted by ESA. We have shown that this system can be considered satisfactorily secure under classic criterions, but some margins for improvement exist particularly regarding the pre-signature and, globally, the possibility to make harder for an opponent the reconstruction of the authentication key. Some simple modifications have been presented in this sense, which leave the generator plant basically unchanged (for example the size of the input and output sequences or the other variables involved) but permit an improvement in performance. In our opinion, they could be easily integrated into the standard, for its updating, taking into account recent developments in the cryptographic techniques.

The proposed modified scheme also exhibits a significant processing time reduction for longer data frames. This is an important factor of merit in the case of satellite applications, most of all if the TC frame lengths will increase in the future by adopting updated CCSDS recommendations (increases of up to 1024 bytes seem realistic and foreseeable).

Moreover, we can expect a further optimization of the new solution in the case of using 32-bit data buses, if and when a technological upgrade of many current satellite systems takes place.

Acknowledgment

The authors wish to thank the anonymous reviewers for having provided many useful comments and suggestions, thus contributing to improve the paper significantly.

References

[1] C.B. Smith and A.F. León, "Authentication in the Telecommand Link to Improve Security," *Proc. TTC 2001*, 2001.

[2] ISO/IEC 9797 Information technology - *Security techniques - Message Authentication Codes (MACs). Part 1: Mechanisms Using a Block Cipher,* 1999, *Part 2: Mechanisms Using a Dedicated Hash-Function*, 2002.

[3] B. Preneel, "Cryptographic Primitives for Information Authentication," *State of the Art and Evolution of Computer Security and Industrial Cryptography*, Lecture Notes in Computer Science 1528, Springer-Verlag, 1998, pp. 50-105.

[4] ESA PSS-04-107, Issue 2, *Packet Telecommand Standard*, ESA, Paris, France, 1992.

[5] ESA PSS-04-151, Issue 1, *Telecommand Decoder Specification*, ESA, Paris, France, 1993.

[6] CCSDS 350.0-G-1, Green Book, Issue 1, *The Application of CCSDS Protocols for Secure Systems*, CCSDS, Washington, D.C., 1999.

[7] W. Stallings, *Cryptography and Network Security: Principles and Practice*, Prentice Hall, 2003.

[8] National Institute of Standards and Technology, "A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications," *NIST Special Publication 800-22*, 15 May 2001.

[9] D.A. Karras and V. Zorkadis, "A Novel Suite of Tests for Evaluating One-Way Hash Functions for Electronic Commerce Applications," *Proc. 26th Euro Micro Conference*, 2000.

[10] N. Ferguson and B. Schneier, *Practical Cryptography*, Wiley, 2003.

[11] http://www.ccsds.org.

[12] Dobbertin, A. Bosselaers, and B. Preneel, "RIPEMD-160: A Strengthened Version of RIPEMD," *Fast Software Encryption, FSE 1996*, Lecture Notes in Computer Science 1039, D. Gollmann (ed.), Springer-Verlag, 1996, pp. 71-82.

[13] A. M. Odlyzko, "The Rise and Fall of Knapsack Cryptosystems," *Cryptology and Computational Number Theory, C. Pomerance (ed.), Am. Math. Soc., Proc. Symp. Appl. Math. #42*, 1990.

Franco Chiaraluce was born in Ancona, Italy, in 1960. He received the Laurea Degree in electronic engineering (summa cum laude) from the University of Ancona in 1985. In 1987, he joined the Department of Electronics and Automatics of the same university. At present, he is an Associate Professor at the Polytechnic University of Marche. His main research interests involve various aspects of communication systems theory and design, with special emphasis on coding, cryptography, and multiple access techniques. He is co-author of more than 150 papers and two books. He is a member of IEICE.

Ennio Gambi is with the Department of Electronics, Artificial Intelligence and Telecommunications of the Polytechnic University of Marche, where he is currently the lecturer for the course of Telecommunication Systems. He is presently working on source and channel coding, encryption, and authentication algorithms, with particular interest in coding systems and transmission of multimedia signals over wired/wireless LAN.

Susanna Spinsante received her "Laurea" degree (summa cum laude) in electronic engineering in 2002 from the University of Ancona, Italy. Since October 2002, she has been a PhD student in electronic engineering and telecommunications at the Polytechnic University of Marche. Since August 2004, she has been a student member of IEEE. From August 2004 to February 2005 she was at the Department of Informatics of the University of Bergen (Norway) as a Marie Curie Fellow, carrying out a 6-month research training at FASTSEC MCTS (Contract no. HPMT-CT-2001-00260) on coding theory and cryptography. Her main research interests are related to security and encryption aspects in communication networks, multimedia applications over IP, coding, and audio/video applications. In the past year she has been involved in research activities on security and authentication issues related to space communications.