An attack on CRC-32 integrity checks of encrypted channels using CBC and CFB modes

Ariel Futoransky
<futo@core-sdi.com>

Emiliano Kargieman
<ek@core-sdi.com>

Ariel M. Pacetti
<ap@core-sdi.com>

CORE SDI S.A.

October, 1998

Keywords: SSH, CRC-32, CBC, CFB.

Abstract

A known-plaintext attack against SSH protocol version 1.5 is described that allows an attacker to insert arbitrary commands in the stream regardless of the authentication protocol used, the block cipher or the key. The attack is based on weakneses of the integrity function used (CRC-32) that become exploitable due to the use of CBC and CFB feedback modes.

1 CFB and CBC feedback modes

We suppose the readers are familiar with the most common modes of operation of symmetric cryptography algorithms, a brief description of two of them follows: Cipher FeedBack (CFB) and Cipher Block Chaining (CBC).
Given a symmetric cipher, the CBC mode of operation is defined as:

$$\begin{aligned} C_0 &= IV \\ C_i &= E_k(P_i xor C_{i-1}) \\ P_i &= E_k^{-1}(C_i) xor C_{i-1} \end{aligned}$$

while CFB is defined as:

$$\begin{aligned} C_0 &= IV \\ C_i &= E_k(C_{i-1}) xor P_i \\ P_i &= E_k(C_{i-1}) xor C_i \end{aligned}$$

Where C_i is the ith block of the ciphertext, P_i is the ith block of the plaintext, $E_k(x)$ is the encryption with the cipher of the block x using the key k and xor is the exclusive or operation. IV is an arbitrary initialization vector.

Normally, feedback modes are used to strengthen a block cipher behaviour by randomizing it's input and reducing the information an eavesdropper can get from repeated plaintext blocks. However, using this feedback modes does not protect the information being transmited against replay or cut-and-paste attacks. For example: if a message consisting of the plaintext blocks $P_1, \ldots, P_m$ is encrypted to the ciphertext blocks $C_1, \ldots, C_m$ using either one of this modes, a malicious attacker with access to the ciphertext can easily launch the following attacks:
a) reduce the length of the message by erasing the last ciphertext blocks.
b) replay some part of the message, appending repeated ciphertext blocks at the end. When the receiver gets the new formed message $C_1, \ldots, C_m$, $C_i, \ldots, C_{i+l}$ and tries to decrypt it he gets a block of gibberish in the $m+1$ place, but from there on $C_{m+2}, \ldots, C_{m+l-i}$ are decrypted correctly to $P_{i+1}, \ldots, P_{i+l}$.

A more dangerous attack can be launched if the attacker knows one pair of plaintext-ciphertext (both C_i and P_i). Here is what she can do.
CBC) append two blocks to the ciphertext blocks:

$$\begin{aligned} C_{m+1} &= X \\ C_{m+2} &= C_i \end{aligned}$$

that will be decrypted to:

$$\begin{aligned} P_{m+1} &= C_j xor E_k^{-1}(X) = ? \\ P_{m+2} &= X xor E_k^{-1}(C_i) = X xor C_{i-1} xor P_i \end{aligned}$$

CFB) append two ciphertext blocks:

$$C_{m+1} = C_{i-1}$$
$$C_{m+2} = X$$

that will be decrypted to:

$$P_{m+1} = C_{i-1} xor E_k(C_j) = ?$$
$$P_{m+2} = X xor E_k(C_{i-1}) = X xor P_i xor C_i$$

In both cases, the attacker cannot control P_{m+1}, but as she knows P_i, C_i and $C_{i=1}$ she can select X so that P_{m+2} results in any block she wants.

This are all attacks against the *integrity* of the message being transmited. The integrity of the message should be assured by the means of a cryptographically secure integrity function (such as one-way hash functions or message authentication codes). However, this cryptographically secure integrity functions are time-consuming and there are implementations that assume that appending a less time-consuming and less secure integrity function to the message before encryption is enough to guarantee the integrity. Among the products that follow this assumption are the widely distributed Secure Shell (SSH) up to protocol version 1.5, some modes of operation of Kerberos and PPTP. To that end, one of the most used integrity functions is CRC-32, originally designed for error detection. In the next section we analyze cyclic redundancy checks and some of their properties.

2 Cyclic redundancy checks

First, we define a bijective mapping between Z_2^n and $Z_2[X]$:

$$\varphi_n : Z_2^n \longrightarrow Z_2[X]$$

$$(a_1, \dots, a_n) \longrightarrow X^m \sum a_i X^{n-i}$$

We'll note with r_m the rest of dividing by the polynomial $m(x)$; and we'll note with J_m the canonical isomorphism between Z_2^m and the polynomials of degree $\leq$ m. $(Z_2^m[X])$

Given $m(x) \in Z_2[X]$ an irreducible polinomial of degree m, we define the *cyclic redundancy check* associated with $m(x)$ (ϕ) as the following composition:

$$Z_2^n \xrightarrow{\varphi_n} Z_2[X] \xrightarrow{r_m} Z_2^m[X] \xrightarrow{J_m^{-1}} Z_2^m$$

$$\phi(a) = J_m^{-1}(r_m(\varphi_n(a)))$$

Let $a = (a_1, \dots, a_n) \in Z_2^n$, $\bar{a} = \phi(a) = (\bar{a}_1, \dots, \bar{a}_m)$. Then for ϕ the following properties hold:

Property 1. *ϕ is a morphism of Z_2 vector spaces.*

Proof. this is derived directly from the definition. (Note that r_m is an aditive morphism).

$\square$

Property 2. $\phi((0, a_2, \ldots, a_n)) = \phi((a_2, \ldots, a_n))$

Proof. $\varphi_n((0, a_2, \ldots, a_n)) = X^m\,(0\,X^n + a_2\,X^{n-1} + \cdots + a_n) = X^m\,(a_2\,X^{n-1} + \cdots + a_n) = \varphi_{n-1}((a_2, \ldots, a_n)) \quad \Rightarrow \quad \phi((0, a_2, \ldots, a_n)) = \phi((a_2, \ldots, a_n))$
$\square$

Property 3. $\phi((a_1, \ldots, a_n, \bar{a}_1, \ldots, \bar{a}_m)) = 0$

Proof.

$$\begin{aligned}
\varphi_{n+m}((a_1, \ldots, a_n, \bar{a}_1, \ldots, \bar{a}_m)) &= X^m\,[\,(\sum a_i\,X^{n-i})X^m + \sum \bar{a}_i\,X^{m-i}] \\
&= X^m\,[m(x)\,h(x) + \sum \bar{a}_i\,X^{m-i} + \sum \bar{a}_i\,X^{m-i}] \\
&= X^m\,m(x)\,h(x) \equiv 0 \pmod{m(x)}
\end{aligned}$$

$\square$

Property 4. $\phi(a_1, \ldots, a_n) \neq 0 \;\Leftrightarrow\; \phi(a_1, \ldots, a_n, 0) \neq 0$

Proof. $m(x)$ is irreducible, then:

$$\begin{aligned}
&\phi(a_1, \ldots, a_n, 0) \;=\; 0 \quad \Leftrightarrow \\
&\quad \varphi_{n+1}(a_1, \ldots, a_n, 0) = [\,X^m\,(\sum a_i\,X^{n-i})\,]X \equiv 0 \pmod{m(x)} \quad \Leftrightarrow \\
&\quad X^m\,(\sum a_i\,X^{n-i}) \equiv 0 \pmod{m(x)} \quad \Leftrightarrow \quad \phi((a_1, \ldots, a_n)) = 0
\end{aligned}$$

$\square$

Definition. Given $X, V \in Z_2^n$ and $M \in Z_2^{m+1}$we define $C_M(X, V) \in Z_2^{(m+1)n}$ as:

$$C_M(X, V) := (A_{1\,1}, \ldots, A_{1\,n}, A_{2\,1}, \ldots\ldots, A_{m+1\,n})$$

where

$$A_{i\,j} = \begin{cases} X_j & \text{if } M_i = 1 \\ V_j & \text{if } M_i = 0 \end{cases}$$

If M := $(M_1, \ldots\,, M_{m+1})$. Then we have that:

$$\begin{aligned}
C_M(X, V) = M_1 * (X, 0, \ldots, 0) + (1 - M_1) * (V, 0, \ldots, 0) + \cdots \\
\cdots + M_{m+1} * (0, \ldots, X) + (1 - M_{m+1}) * (0, \ldots, V)
\end{aligned}$$

Proposition 1. *given m(x) an irreducible polynomial in* $Z_2[x]$ *, of degree m ,* $\exists\ M \in Z_2^{m+1}$ *such that:* $\phi(C_M(X, V)) = \phi(C_M(0, V))\ \forall X \in Z_2^n\ ,\ \forall V \in Z_2^n$

Proof. Property 1 tells us that ϕ is a morphism so we have:

$$\begin{aligned} \phi(C_M(X,V)) = \phi(C_M(0,V)) \quad &\Leftrightarrow \quad \phi(C_M(X,V) - C_M(0,V)) = 0 \\ &\Leftrightarrow \quad \phi(C_M(X,V) + C_M(0,V)) = 0 \end{aligned}$$

(we are working over Z_2)

We must see that $C_M(X,V) + C_M(0,V) \in \mathrm{Ker}(\phi)$.
For $X, V \in Z_2^n$, we are going to write (X,V) for $(X_1, \ldots, X_n, V_1, \ldots, V_n)$.
If $M = (M_1, \ldots, M_{m+1})$,we have that:

$$\begin{aligned} &C_M(X,V) = M_1 * (X,0,\ldots,0) + (1-M_1) * (V,0,\ldots,0) + \cdots \\ &\qquad \cdots + M_{m+1} * (0,\ldots,X) + (1-M_{m+1}) * (0,\ldots,V) \quad \Rightarrow \\ &C_M(X,V) + C_M(0,V) = M_1 * (X,0,\ldots,0) + \cdots + M_{m+1} * (0,\ldots,X) = \\ &= (M_1X, \ldots, M_{m+1}X) \end{aligned}$$

So, what we have to prove is that $\exists$ M such that $\phi(M_1X, \ldots, M_{m+1}X) = 0$, $\forall X$ (Note that this doesn't depend on V)
It's enough to prove it in a base of Z_2^n , because ϕ is a morphism.
Let $\{e_i\}$ be the canonical base.

$$\begin{aligned} \phi(M_1e_i, \ldots, M_{m+1}e_i) = 0 \ &\Leftrightarrow \ r_m(\varphi_n(M_1e_i, \ldots, M_{m+1}e_i) = 0 \\ &\Leftrightarrow \ m(x) \setminus \varphi_n(M_1e_i, \ldots, M_{m+1}e_i) \end{aligned}$$

But

$$\begin{aligned} \varphi_n(M_1e_i, \ldots, M_{m+1}e_i) &= x^m * (M_1x^{(m+1)n-i} + M_2x^{mn-i} + \cdots + M_{m+1}x^{n-i}) = \\ &= x^{m+n-i} * (M_1x^{mn} + M_2x^{(m-1)n} + \cdots + M_mx^n + M_{m+1}) \end{aligned}$$

Then, as $m(x)$ is irreducible, it's enough to find M such that

$$m(x) \setminus (M_1x^{mn} + M_2x^{(m-1)n} + \cdots + M_mx^n + M_{m+1}).$$

By Euclid's algorithm:

$$(M_1x^{mn} + M_2x^{(m-1)n} + \cdots + M_mx^n + M_{m+1}) = m * q + r \quad \text{with} \quad deg(r) \leq m-1$$

Asking $r = 0$ we have an homogeneous linear system with $m+1$ variables $(M_i)$and $deg(r)$ equations. Such a system has a non-trivial solution. $\square$

Corolary. $\phi(C_M(X,V)) = \phi(C_M(0,V))\ \forall X, V \in Z_2^n \Leftrightarrow \phi(C_M(X,V), 0, \ldots, 0) = \phi(C_M(0,V), 0, \ldots, 0)\ \forall X, V \in Z_2^n$.

Proof. Let k be the number of zeros in the expresion above.

$$\begin{aligned}
&\phi(C_M(X,V),0,\dots,0) = \phi(C_M(0,V),0,\dots,0) \Leftrightarrow \\
\Leftrightarrow\quad & \phi((C_M(X,V),0,\dots,0) + (C_M(0,V),0,\dots,0)) = 0 \\
\Leftrightarrow\quad & \phi((C_M(X,V) + C_M(0,V),0,\dots,0) = 0 \\
\Leftrightarrow\quad & m(x) \setminus x^m\ \varphi_n(C_M(X,V) + C_M(0,V))\, x^k \\
\Leftrightarrow\quad & m(x) \setminus x^{m+k}\ \varphi_n(C_M(X,V) + C_M(0,V)) \\
\Leftrightarrow\quad & m(x) \setminus \varphi_n(C_M(X,V) + C_M(0,V)) \\
\Leftrightarrow\quad & \phi((C_M(X,V) + (C_M(0,V)) = 0 \quad \Leftrightarrow \quad \phi(C_M(X,V)) = \phi(C_M(0,V))
\end{aligned}$$

□

Proposition 2. *If* $C, D \in Z_2^n$ *and* $\phi(C) \neq \phi(D) \Rightarrow \forall A \in Z_2^m\ \exists M \in Z_2^m$ *such that* $\phi(C_M(C,D),C,C) = A$ *if* $n = 2^r$.

Proof. Let's call:

$$\begin{aligned}
\Gamma : Z_2^m \quad &\longrightarrow \quad Z_2^m \\
(a_1,\dots,a_m) \quad &\longrightarrow \quad \phi(a_1\, C + (1+a_1)\, D, \dots, a_m\, C + (1+a_m)\, D) \\
\Gamma \ &= \ \phi(C_{(a_1,\dots,a_m)}(C,D))
\end{aligned}$$

$$\phi(C_M(C,D),C,C) \ = \ \phi(C_M(C,D),0,\dots,0) \ + \ \phi(0,\dots,0,C,C)$$

Then it's enough to prove that Γ is biyective. But $\# Z_2^m \ = \ \# Z_2^m \ < \infty$ then all we have to prove is that Γ is inyective.

$$\begin{aligned}
&\Gamma(a_1,\dots,a_m) \ = \ \Gamma(b_1,\dots,b_m) \quad \Leftrightarrow \\
&\phi(a_1\, C + (1+a_1)\, D, \dots, a_m\, C + (1+a_m)\, D) \ = \ \phi(b_1\, C + (1+b_1)\, D, \dots, b_m\, C + (1+b_m)\, D) \\
\Leftrightarrow\quad & \phi((a_1+b_1)\, C + (a_1+b_1)\, D, \dots, (a_m+b_m)\, C + (a_m+b_m)\, D) \ = \ 0 \\
\Leftrightarrow\quad & \phi((a_1+b_1)\,(C+D), \dots, (a_m+b_m)\,(C+D)) \ = \ 0
\end{aligned}$$

Let's call $h_i \ = \ a_i + b_i$. Then:

$$\begin{aligned}
&\Gamma(a_1,\dots,a_m) \ = \ \Gamma(b_1,\dots,b_m) \quad \Leftrightarrow \\
\Leftrightarrow\quad & m(x) \setminus x^m\ \varphi_n(C+D)\, (h_1\, x^{n\,(m-1)} + h_2\, x^{n\,(m-2)} + \cdots + h_{m-1}\, x^n + h_m)
\end{aligned}$$

Now we 're going to use that $n = 2^r$, because in Z_2 we have:

$$(a+b)^2 \ = \ a^2 + b^2$$

Square is an isomorphism in Z_2 spaces.Then:

$$(h_1\, x^{2^r\,(m-1)} + h_2\, x^{2^r\,(m-2)} + \cdots + h_{m-1}\, x^{2^r} + h_m) \ = \ (h_1\, x^{(m-1)} + h_2\, x^{(m-2)} + \cdots + h_{m-1}\, x + h_m)^{2^r}$$

Using that $m(x)$ is irreducible , we have that:

$$m(x) \setminus (h_1\, x^{(m-1)} + h_2\, x^{(m-2)} + \cdots + h_{m-1}\, x + h_m)$$

Which is a polynomial of degree $<$ deg$(m(x))\Rightarrow h_i = 0\,\forall i \Rightarrow a_i = b_i\,\forall i$

$\square$

Note. the last proposition is not valid if $n \neq 2^r$.

Let's take $m(x) = x^6 + x^3 + 1$ which is irreducible in $Z_2[x]$.

Then taking n = 3 we have that:

$$\phi(C,D,D,C,D,C) = \phi(C,D,D,D,C,D) \quad \Leftrightarrow \quad \phi(0,0,0,C{+}D,C{+}D,C{+}D) = 0 \quad \Leftrightarrow$$

$$\Leftrightarrow \quad m(x) \setminus x^6\, \varphi_n(C{+}D)\,(x^6{+}x^3{+}1) \quad \text{what means that } \Gamma \text{ is not inyective.}$$

Now we need to find a way to construct the mask M.

Proposition 3. $\Gamma(a_1+b_1,\ldots,a_m+b_m) = \Gamma(a_1,\ldots,a_m) + \Gamma(b_1+\ldots+b_m) + \Gamma(0,\ldots,0)$.

Proof.

$$\begin{aligned}
\Gamma(a_1+b_1,\ldots,a_m+b_m) &= \phi((a_1+b_1)\,C + (1+a_1+b_1)\,D,\ldots,(a_m+b_m)\,C + (1+a_m+b_m)D)\\
&= \phi((a_1+b_1)\,C + (a_1+b_1)\,D,\ldots,(a_m+b_m)\,C + (a_m+b_m)\,D) + \phi(D,\ldots,D)\\
&= \Gamma(a_1,\ldots,a_m) + \Gamma(b_1,\ldots,b_m) + \Gamma(0,\ldots,0)
\end{aligned}$$

$\square$

Construction.

Let $p \in Z_2^m$ *and* $l \in Z_2^m$ *such that* $\Gamma(l_1,\ldots,l_m) = (p_1,\ldots,p_m)$

$$\begin{aligned}
&\Rightarrow \Gamma(l_1,\ldots,l_m) = \Gamma(l_1,0,\ldots,0) + \Gamma(0,l_2,\ldots,l_m) + \Gamma(0,\ldots,0) =\\
&= l_1\Gamma(e_1) + l_1\Gamma(0) + \Gamma(0,l_2,\ldots,l_m) =\\
&= l_1\,\Gamma(e_1) + \cdots + l_m\,\Gamma(e_m) + (l_1+\cdots+l_m)\,\Gamma(0)
\end{aligned}$$

Where e_i *are the elements of the canonical base of* Z_2^m.
Calling $a_{m+1} = l_1 + \cdots + l_m$ *, we have to solve the following system:*

$$\begin{cases} l_1\,\Gamma(e_1) + \cdots + a_m\,\Gamma(e_m) + a_{m+1}\,\Gamma(0) = p \\ a_1 + \cdots + a_m + a_{m+1} = 0 \end{cases}$$

Which has an unique solution because Γ *is biyective.*

3 A known-plaintext attack on SSH protocol version 1.5

The SSH protocol, up to version 1.5, used a CRC-32 appended to a packet before encryption to check for integrity. CRC-32 is the 32-bits cyclic redundancy check associated with the polynomial $m(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^{8} + x^{7} + x^{5} + x^{4} + x^{2} + x^{1} + 1$.
From here on, we assume that the packets are encrypted using CFB or CBC modes with a 64-bit block cipher, as in the DES-CBC and IDEA-CFB standard modes of SSH. The attack will take place after both ends have been authenticated and a session key negotiated.

3.1 Description of an SSH packet

A packet is formed by the following fields: a 4-bytes data length field, 1 to 8 bytes of padding, a 1-byte type field, the data and a 4-byte CRC-32 field.

The data length field is unencrypted, while the padding, type, data and CRC-32 fields are encrypted. The last block of the previous packet is used as IV for the feedback, in the first packet of a connection the IV is set to zero. The padding is added to make the packet size be a multiple of 64-bits, and to include at least one padding byte (i.e. if the original packet is a multiple of 64 bits, then 64 bits of padding are prepended to the message). The CRC-32 is calculated over the padding, type and data fields.
The data field structure depends on the type of the message, there are two type of messages that are of a particular interest for us and can be directly used for the attack: SSH_STDIN_DATA and SSH_STDOUT_DATA. Both this message types define the following data structure:

As the size of the ciphertext blocks is 64-bits, within 2 ciphertext blocks we can include:

+ 8 bytes of padding
+ 1 byte type (SSH_STDIN_DATA or SSH_STDOUT_DATA)
+ 4 bytes message len
+ 3 bytes of message data

SSH will choose to close the connection in one of this three cases:
- The CRC field does not match the CRC of the decrypted packet.
- The 32-bit message length $\neq$ data length$-$length of padding-1
- The message type is invalid

3.2 The attack

3.2.1 CBC mode

The attacker needs two known plaintext/ciphertext blocks (C_i,P_i and C_j,P_j). Then the following steps can be used to forge a valid packet into the encrypted stream:

I The attacker chooses the packet length to be a multiple of 64-bits, so the padding will be 64-bits (1 ciphertext block).

II She starts the packet with two ciphertext blocks:

$$C_1 = X$$
$$C_2 = C_i$$

Where X is choosen to make P_2 a block with a valid type, a valid message length and 3 bytes of selected data. P_l will be unknown to the attacker, but it will fit into the padding field of the packet, and it will not used in the protocol (except for the CRC-32 computation).

III The attacker uses the mask M from Proposition 1 to append 32 64-bits ciphertext blocks ($C_3, \ldots, C_{34}$), each block either X or C_i, to make the final CRC-32 of the decrypted packet independent of the value of $E_k^{-1}(X)$.

IV The attacker appends 32 64-bits blocks to the cipher stream ($C_{35}, \ldots, C_{66}$), each block either C_i or C_j, to fix the CRC-32 result to the value $P_i xor C_i$ (as is described in proposition 2 and the construction at the end of the previous section). Note that for proposition 2 to be valid we need that $\phi(C_i) \neq phi(C_j)$, but the probability of $\phi(C_i)$ being equal to $\phi(C_j)$ is small enough to be ignored ($\leq frac12^{32}$).

V The attacker appends two C_i blocks to the cipherstream:

$$C_{67} = C_i$$
$$C_{68} = C_i$$

This packet has a valid type, a valid message length and a valid CRC-32 field, and the first 3 bytes of data in the packet can be selected by the attacker. Moreover this procedure can be repeated as many times as the attacker wants.

3.2.2 CFB-64 mode

The attacker needs only one pair of known plaintext/ciphertext blocks (C_i, P_i), she needs to know C_{i+1} and needs to know the last ciphertext block of the last transmited packet which is going to be used as the initialization vector for the feedback (we will call this ciphertext block I). To forge a valid packet, the attacker can follow the following steps:

I The attacker chooses the packet length to be a multiple of 64-bits, so there will be 64-bits of padding.

II She starts the packet with two ciphertext blocks:

$$C_1 = C_i$$
$$C_2 = X$$

The attacker selects X to make P_2 have a valid type, message length and 3 bytes of selected data. P_1 is unknown to the attacker, but is entirely contained in the padding field.

III The attacker uses the mask M from proposition 1 to append 32 64-bit ciphertext blocks, each block either I or C_i, defining a particular pattern that will make the final CRC-32 independent of $E_k(I)$.

IV The attacker uses the mask M to append 32 64-bit ciphertext blocks, each block either C_i or X, so that the CRC-32 of the decrypted packet does not depend on the value of $E_k^{-1}(X)$.

V The attacker appends two ciphertext blocks: $C_{67} = C_i$ and $C_{68} = Y$. Y is selected so that Y xor C_{i+1} equals the CRC-32 of the constructed packet. The CRC-32 can be calculated because it only depends on I and P_i both of which are known to the attacker.

References

[1] T. Ritter *The great CRC mystery* , Dr. Dobb's Journal of software tools , (1986) February 11(2), 26-34, 76-83.

[2] T. Ylonen *The SSH (Secure Shell) Remote Login Protocol*, Helsinki University of Technology. November 15th 1995 (draft expired on May 15th, 1996)Included as the file ./RFC in the ssh distribution.

[3] T. Ylonen, T. Kivinen, M. Saarinen *SSH Protocol Architecture*, draft-ietf-secsh-architecture-01.txt.gz, November 7th, 1997.

[4] T. Ylonen, T. Kivinen, M. Saarinen *SSH Connection Protocol*, draft-ietf-secsh-connect-03.txt.gz, November 7th, 1997.

[5] T. Ylonen, T. Kivinen, M. Saarinen *SSH Authentication Protocol*, draft-ietf-secsh-userauth-03.txt.gz, November 7th, 1997.

[6] T. Ylonen, T. Kivinen, M. Saarinen *SSH Transport Layer Protocol*, draft-ietf-secsh-transport-03.txt.gz, November 7th, 1997.
(drafts expired on May 7th, 1998) All Internet drafts are available at <ftp://ftp.isi.edu/internet-drafts/>

[7] L. Joncheray *Simple Active Attack Against TCP*, Merit Networks Inc., 5th USENIX Security Simposium. 1995.

[8] S. Bellovin *Problem areas for the IP Security Protocols* USENIX 1996. <http://www.research.att.com/ smb/papers/badesp.ps>

[9] S. Stubblebine, V. Gligor *On Message Integrity in Cryptographic Protocols*, IEEE Computer Society Symposium on Research in Security and Privacy, Oakland, CA, May, 1992, pp. 85-104.

[10] S. Stubblebine, V. Gligor *Protocol Design for Integrity Protection*, IEEE Computer Society Symposium on Research in Security and Privacy, Oakland, CA, May, 1993, pp. 41-53.

[11] CORE SDI S.A. *SSH insertion attack*[1] , Security advisory, <http://www.core-sdi.com>, 1998.

[1]Special thanks go to David Wagner for providing us with a lot of useful references to previous related work.