

# Review: Plaintext Recovery Attacks against SSH

## Introduction to Cryptographic Algorithms '12/'13

Raoul Estourgie  
Ben Brücker

Institute for Computing and Information Sciences  
Radboud University Nijmegen

# Outline

## Introduction

What is SSH?

What is the SSH-BPP protocol?

Open SSH implementation of SSH BPP

## The attack

Some notations

Stage 1: Recovering first 14 bits of plaintext

Stage 2: Some more bits

Stage 3: Recovering 32 bits of plaintext

## Countermeasures

## Conclusion

## Questions

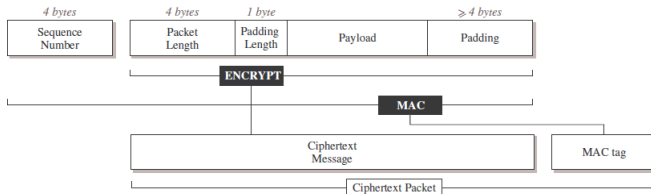
# What is SSH?

- Secure Shell (SSH) connects computers securely over insecure network connections.
- It was released in 1995 and was designed to replace rlogin, rsh, Telnet and similar insecure protocols.
- The SSH protocol covers authentication, confidentiality and integrity.
- Our review article "Plaintext Recovery Attacks against SSH" paper focuses on the OpenSSH implementation.

# What is the SSH-BPP protocol?

- The Binary Packet Protocol (BPP) of SSH encrypts a plaintext and then protects its integrity by appending a MAC value.
- Prefixed with 4 byte packet-length, 1 byte padding-length
- Suffixed with 4 to 255 bytes of padding
- The message is then encrypted with a cypher of choice, for example aes128-cbc.
- MAC is calculated over this message and a 32-bit packet sequence number
- MAC is appended to the message

# Schematic of a BPP block



## What is the SSH-BPP protocol? cont.

- The packets then form a data stream since the encryption is in CBC mode.
- Every packet  $i - 1$  on a connection will be the initialization vector (IV) for packet  $i$  on the same connection.
- For decryption it is essential that the receiver decrypts the first ciphertext block to be able to read the length field.
- The SSH protocol also specifies error handling for the BPP protocol.
- Terminate whenever a transmission error occurs or MAC verification fails.
- Implementations are free to send error messages to their peer when an error occurs.

# What is the SSH-BPP protocol?

- Open SSH first performs a length check. If the length given in the length field is not between 5 and  $2^{18}$  it sends a *SSH2 MSG DISCONNECTED* error back to the sender.
- Next OpenSSH checks that the total number of bytes expected is indeed a multiple of the block size. When this check fails, the TCP connection will terminate without an error message.
- When all data for the package has arrived, OpenSSH performs a MAC check. If this check fails, a *Corrupted MAC on input.* error message is returned to the sender.
- Note that an attacker can differentiate those failure modes.

## Some notations

### Key

$K$  is the key of our block cipher

### Encrypt/Decrypt

$F_k$  and  $F_k^{-1}$  are the encryption and decryption operations of the block cipher

### CBC Mode

Given a sequence  $p_1, p_2, \dots, p_n$  of plaintext blocks making up a packet, we have:  $c_i = F_k(c_{i-1} \oplus p_i), i = 1, 2, \dots, n$  where  $c_0$ , the Initializing Vector (IV), is the last block of the previous ciphertext. Decryption works as follows:  $p_i = c_{i-1} \oplus F_k^{-1}(c_i), i = 1, 2, \dots, n$



## Recovering first 14 bits of plaintext

- OpenSSH implementation of BPP only supports lengths up to  $2^{18}$  bits.
- Length field is 4 bytes long.
- A packet will only pass the length check if the first  $32 - 18 = 14$  bits are all 0
- So when no *SSH2 MSG DISCONNECTED* error occurs, we recovered first 14 bits of plaintext.

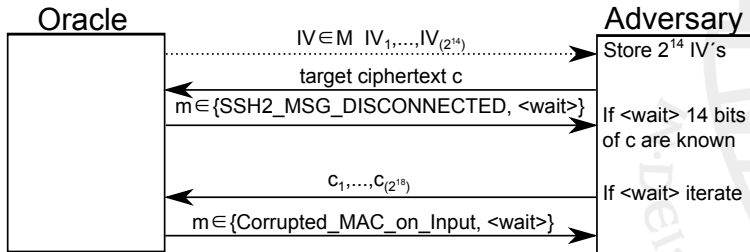
## Recovering some more bits

- When the ciphertext block passes the block length check we get to know more bits
- In case of  $L = 16$  (as with AES) the we know the last 4 bits for a total of 18 bits.
- In case of  $L = 8$  (as with 3des) the we know the last 3 bits for a total of 17 bits.

## Recovering 32 bits of plaintext

- When the first 2 checks succeed, the SSH connection enters wait state.
- Feed random ciphertext blocks into this connection and wait after each block.
- When the target returns a *Corrupted MAC on input*. error we know that it received enough blocks to trigger a MAC check.
- At this point we know exactly how long the packet length is, and therefore all 32 bits of the length-field.
- Because of the chaining property of the CBC mode these 32-bits leak information about the rest of the ciphertext.

# Security game



# Countermeasures

- Return the same error message when either the length check or the block-length check failed.
- Randomize the length field if the length check fails.

## Conclusion

It is possible to recover plaintext bits from a proven secure SSH implementation. Therefore it is also hard to know whether improvements to resolve this issue won't lead to new attacks on these SSH implementations.

# Questions

Questions?

