SOLUTION NOTES

Security (Part II) 2002 Paper 8 Question 6 (MGK)

(a) The Trusted Computing Base are those parts of a system that enforce a security policy and whose correct operation is sufficient for successful enforcement. Under Unix, access control decisions are made inside the kernel, which operates in a separate address space from user processes and whose code and data structures are protected from tampering by normal user processes. The TCB therefore includes all parts of the kernel, including its device drivers. All processes run by the root user can not only by-pass access control completely, special device drivers even allow them direct access to the kernel address space. As a result, all these, as well as stored programs with setuid root, are part of the TCB as well, as are the libraries and development tools used to build them. The TCB is not restricted to software alone. Mass storage devices and network file servers are part of the TCB if they store software that is within the TCB, such as programs executed with root privileges. The CPU and other hardware components are part of the TCB as well, as the kernel protection relies on their correct operation.

(b) CRCs are linear codes. An attacker who wants to modify selected bits at known positions in a CRC protected packet can generate a packet of equal length with a one bit at each bit position that is to be modified. After calculating a CRC field for this new packet and XORing the two packets together, the resulting packet will also have a correct CRC field plus the desired bit modifications. Like any stream cipher, counter mode is linear, therefore XORing a value to the ciphertext corresponds to XORing the same value onto the plaintext. As a result, an attacker can easily toggle any bit position in a packet, such as for example the bit that signals whether the entered PIN was correct or whether there are sufficient funds in the account.

(c) (i) The phone sends a nonce N, and receives as a reply the card's serial number S, the current card value V, as well as a hash of the shared secret K, N, S, and V. This confirms to the phone that there is a card that knows the shared secret and has a certain stored value. The phone then requests the card to decrement its value by the call cost C. The card replies with an update of the hash that shows the new card value. This hash still contains the same nonce and serial number, which shows that it is really the same card in the same transaction that has now a decremented value, not a second card with a lower value that has been fed the same nonce.

$$\begin{aligned} \text{phone} \rightarrow \text{card} &: N \\ \text{card} \rightarrow \text{phone} &: S, V, H(K, N, S, V) \\ \text{phone} \rightarrow \text{card} &: C \\ \text{card} \rightarrow \text{phone} &: H(K, N, S, V - C) \end{aligned}$$

(ii) If a single phone card gets successfully reverse engineered and K becomes known to the attacker, there is no secret left in any of the genuine cards to distinguish

them from illicit card emulators. A solution is to replace the common key K with a card-specific key K_S, that can be calculated by the phone with the help of a masterkey K_M: $K_S = H(K_M, S)$. This authenticates the card serial number and allows the use of a blacklist with the serial numbers of compromised cards.

[*This question refers mainly to the course chapters on "trusted computing base", "Unix security", "block ciphers", and "authentication and key exchange".*]