Software Engineering Programme
Software and Systems Security

assessment

student	Mayur Pant
course	Security Principles (SPR)
date	January 2013
grade	75

report

First, we must apologise that these reports are returning to you over a month later than planned. We do appreciate the problems this causes in planning and expectations, and will strive to do better in future.

Your answer is long – too long, really – and very thorough, although the organisation of the ideas is sometimes opaque. It has quite a few tangential ideas, but overall it shows an excellent grasp of the topics of the course. The version which reached me had all the ligatures (fi, ti, fl, etc.) missing: this made it incredibly difficult to read! If you have an opportunity to verify your PDF on a different machine in future, please do.

In Question 1, your taxonomy of threats and vulnerabilities isn't quite right: "Coins should be resilient to unauthorized replication" is a security requirement, not a vulnerability. The vulnerability would be more along the lines of "As stand-alone digital tokens, DigiPounds are inherently vulnerable to duplication." However, your intent is clear, and the analysis is thorough overall.

In Question 2, having both a keyed MAC *and* a digital signature seems redundant. Moreover, it is not clear how the symmetric key to be used in the MAC is to distributed/managed. In Section 6 you make mention of this being a session key, but the connection is not followed up, and is non-obvious.

Your answer to Question 3 is fine – though sticking to the form of notation we used in class would have been helpful. The choice of SSL is reasonable: an alternative would be the Lowe-modified Needham-Schroeder.

In Question 4, you go into considerable detail on the selection of ciphers and hash functions. This appears sound, but is probably more than was needed here. I was amused at the line "... even Microsoft System Center Configuration Manager 2012 uses 3DES for password protecting ...". I have much respect for Microsoft's security engineers, but this is not really proof of anything. Typically, most applications would use AES-192, say, instead of 3DES today, because it offers a similar strength and significant performance benefits. You mention some current good choices of hash function: SHA-3 is poised to become the widespread standard, one imagines. Pragmatically, at this level of design we should simply refer to "a good symmetric cipher", "a good asymmetric algorithm", etc.: if more detail is needed on the selection of algorithms and key sizes, it would be wise to consult a specialist.

In Question 5, you present several solutions – without obviously using the protocol notation explored in class. The temptation to emulate BitCoin is wisely resisted – the design of DigiPound is deliberately different. Of course, this does not preclude using the bank – or a third-party “wallet keeper” to keep track of coin ownership (in a privacy-preserving way). You also quote the approach of Chaum, as told by Schneier, which is fine, but probably over-the-top, in truth.

Your answer to Q6 covers a range of possible attacks – some theoretical and some all too realistic. The protocol-based attacks *should* be prevented by the choice of protocols in answers 3, 4, and 5.