Anonymous Digital Cash Protocols

by David G. Messerschmitt

Supplementary section for Understanding Networked Applications: A First Course, Morgan Kaufmann, 1999.

Digital cash protocols that achieve spending anonymity are quite complicated, especially if they are to provide only conditional anonymity for the first (legitimate) spending. It is beyond the scope of this book to describe such a protocol in detail. However, three key ideas (all attributed to David Chaum) will illustrate some clever ideas behind such protocols.

Recall that the objective is for the consumer to obtain a token of value from the issuer, that token containing a unique identifier which the issuer does not see until after the cash is spent. The first idea is to let the consumer (rather than the issuer) create each token and send it to the issuer for signing. If each identifier is chosen randomly from a large number of alternatives, two tokens having the same identifier is unlikely.

The second idea is to hide the identifier from the issuer. The *blind signature* allows the issuer to sign a token of value, while preventing it from seeing the contents.

Analogy: The physical analogy to the blind signature is for a consumer to create a paper cash token, place it in an envelope with a piece of carbon paper, and give it to the bank for authentication. She also authorizes the bank to deduct the amount of the cash token from her account. The bank authenticates the cash token by embossing the outside of the envelope. Because of the carbon paper, the bank's embossing also carries through to the token within. When returned to the consumer, she opens the envelope and removes the token, which has been authenticated by the bank's embossing.

Such a blind signature protocol can be based, for example, on RSA encryption (see the book homepage for a derivation of this). Recall that the digital signature is the message digest (MD) of the digital cash token encrypted by the issuer's secret key. The idea is for the consumer to generate a random blinding factor B, encrypt it using the issuer's public key, and then multiply MD by the encrypted blinding factor, yielding a blinded MD that is sent to the issuer. When the issuer returns that blinded MD encrypted by the issuer's secret key, the consumer can remove the blinding factor by dividing by B. The principle behind the protocol is that B encrypted by the issuer's public key followed by secret key results in B, which the consumer knows but the issuer doesn't.

As described thus far, the consumer could put any valuation on the token she wants. The issuer will be reluctant to sign a token without being able to see its value. However, without even seeing the token the issuer can be assured of consumer honesty by a *cut-and-choose protocol.*

Analogy: Suppose you cut a piece of pie in half to share it with your sibling. A cut-and-choose protocol works like this (no doubt you have actually done this): You do the cut (nominally in half) and allow your sibling to choose either piece. The point of this protocol is that it discourages greed. If you deliberately make one piece larger, you actually end up with less, because your sibling will surely choose the larger piece.

The cut-and-choose protocol for digital cash works the same way. The consumer, rather than generating just one instance of a digital cash token to be signed by the bank, actually generates n

tokens, each with the same value and each with a different random identifier. The consumer calculates the MD for each of the n tokens, and sends them all (appropriately blinded) to the issuer, which gets to choose which of the n MD's it will encrypt and return to the consumer. The issuer requests and is provided the remaining $n-1$ tokens, which it validates as having the correct MD's, the proper value, and distinct identifiers. To successfully cheat, the consumer would have to correctly guess in advance the MD the issuer will choose, and modify the value on that one and only that one. A sufficiently large fine for cheating will render this strategy financially ineffective.

Concepts

- Blind signature
- Cut-and-choose protocol

Exercises

E1. Explain how the SET dual signature prevents the following forms of cheating. Who can detect the cheating, and how would they prove it in court?

a. The customer later claims that he did not order the merchandise, even though his credit card was charged.

b. The customer later claims that he only offered to pay $10, when in fact the original offer was $15.

c. The customer admits to making an offer, but denies having charged it to his credit card.

E2. Consider the following blind signature protocol for digital cash: The consumer calculates the message digest (MD) of a digital cash token and sends it to the issuer. The issuer signs it and returns it to the consumer. Since the issuer cannot recover the token from the MD, it cannot infer the token identifier. That being the case, why does this protocol fail to protect consumer privacy?

E3. As a seasoned criminal, you plot to extort a large sum of money from a wealthy individual by kidnaping their son for ransom. Compare the relative merits of digital cash with spending anonymity and hard cash as ransom from the perspective of both the criminal and the law enforcement authorities.

E4. Suppose digital cash comes in the same denominations as U.S. coins: 1, 5, 10, 25, 50, and 100 cents. How large must a determined cheater in the cut-and-choose protocol be fined to make cheating unprofitable? Assume the fine for cheating is the same regardless of the denominations involved, and assume that $n = 100$.

E5. Devise an approach using the blind signature and cut-and-choose protocol that would absolutely prevent the merchant and acquirer banks to collude to link information about the order and payment authorization in a modification to the SET protocol.