Signet: Low-cost Auditable Transactions Using SIMs and Mobile Phones

Michael Paik
New York University
mpaik@cs.nyu.edu

Lakshminarayanan Subramanian
New York University
lakshmi@cs.nyu.edu

ABSTRACT

The absence of reliable network connectivity in the developing world has resulted in the use of paper receipts remaining the de facto standard for tracking transactions of various types. This includes both cash transactions (microfinance-related disbursements and repayments, purchases, money transfers) and noncash goods (food commodities to/from godowns or warehouses).

Such receipts are susceptible to loss, damage and alteration, with the last in particular severely compromising systems which depend on quotas or disbursement and repayment. Similarly, they allow go-betweens and agents to falsify payment or disbursement information and embezzle or misdirect funds or goods.

This paper describes Signet, a system which uses the computational power of commodity mobile telephones and security enabled Subscriber Identity Module (SIM) cards to create a secure and auditable record for atomic in-person transactions in the developing world. Signet performs this function at very low operating cost and without requiring continuous connectivity to a trusted third party.

Categories and Subject Descriptors

K.4.f [**Computing Milieux**]: Computers and Society—*Security*; H.1.2 [**Information Systems**]: User/Machine Systems—*Human Factors*; J.9.d [**Computer Applications**]: Mobile Applications—*Pervasive Computing*

General Terms

Human Factors, Security, Economics, Design

Keywords

Transactions, Signing, Audit, Fair Exchange

SOSP NSDR 2009 Big Sky, Montana

1. INTRODUCTION

The delivery of various services requires the ability to accurately and securely track when funds, goods, or other resources associated with these services have changed hands [23]. For many purposes, handwritten or machine-printed receipts have served in this role until the present day, and these have been fraught with various security shortcomings.

Paper receipts, much like paper currency, are susceptible to destruction, loss, and falsification / forgery. However, without the same standardization and perceived intrinsic value that currency has enjoyed for centuries in most milieux, each of these threats is much greater in the case of receipts. The danger of forgery, in particular, is made even more egregious by the fact that receipts typically bear arbitrary amounts, which can be entered at will.

Moreover, in contexts in the developing world with limited literacy, signatures are rudimentary at best and typically easily forged[1] - sometimes being as simple as an 'X', a fact which has held as true in 19th century America [22] as in 20th century Rajasthan [24], India.

The Reserve Bank of India notes these same concerns in a whitepaper on construction of a national Warehouse Receipt system: "Paper Warehouse Receipts suffer from various shortcomings such as difficulty in splitting, risk of forgery, risk of theft / mutilation, etc. The [sic] electronic Warehouse Receipts remove such shortcomings and provide for faster movement of information and automatic creation of audit trail." [2]

As high-speed network connectivity is sparse in many of the rural regions most served by microfinance institutions (MFIs) and warehouse/godown receipt systems, solving the problem of providing digital, auditable receipts to each participant in these systems is nontrivial.

1.1 Two Straw Men

One obvious, if suboptimal, approach to providing electronic receipts to parties of the aforementioned types of transactions is to create network infrastructure servicing every point at which such a transaction might be made. Standard e-commerce techniques would then suffice to provide a secure transaction with an auditable, centrally stored record. However, the cost of establishing such a network is clearly prohibitive in rural and sparsely populated areas with existing commercial technology. In addition, if the central store

[1]Kaestle [18] finds that signatures are indicative of those people who are "minimally literate" and by the contrapositive people who are illiterate cannot be relied upon to produce meaningful signatures.

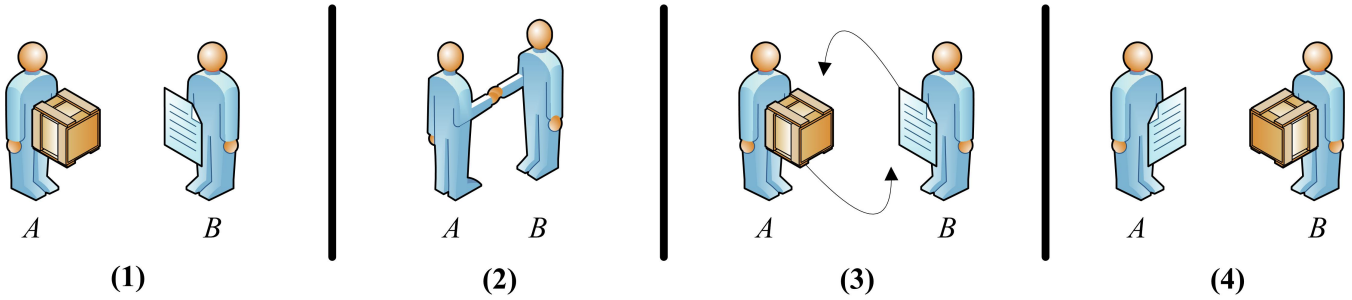

Figure 1: A receipt transaction. Party A wishes to exchange goods or funds g for a receipt r from B (1) and the two parties come to an agreement (2). The r containing some metadata $desc_g$ and B's signature $sign_B(A, desc_g)$ is exchanged for g (3), and the two parties take possession (4).

is not trusted (e.g. part of a corrupt regime or susceptible to falsification for embezzlement), the possibility remains of one party to the transaction creating a forged receipt.

This being the case, it is imperative that both parties to any transaction receive some verifiable, auditable receipt of said transaction, whether or not a third party exists to keep an independent record. Additionally any solution should have a low enough cost to allow each participant in the system to have some digital store and have the system remain financially viable.

Another approach is to use mobile telephones and conduct transactions over SMS, USSD or GPRS, exchanging information between the telephone of each party and the central server, in a manner analogous to the operation of systems like M-PESA [9]. This solution has several disadvantages. Firstly, it requires each party to have a mobile phone and phone credit when performing any transaction, both of which are high hurdles to the very poor. While some MFIs include the cost of a phone in their initial disbursement of cash in a microloan [26] (or whose disbursement consists entirely of the price of a mobile phone and airtime [11]), this is not universally the case, and no such provision exists for the rural poor farmer. Additionally, any such transaction requires both cellular coverage and the successful in-order delivery of the various handshake and payload messages involved in securing such a transaction. With respect to the former condition, while cellular coverage is good in many places, it is far from entirely ubiquitous in rural areas. Regarding the latter, it is not guaranteed when using cellular services that data-bearing messages will be delivered in order, or, in fact, at all. Finally there is the per-transaction marginal cost of those messages, assuming they are not subsidized by the carrier.

1.2 The Role of Signet

We attempt to circumvent the weaknesses outlined above in the design of Signet. While we wish to make use of existing infrastructure capacity and well-understood technology metaphors in the contexts of interest, we do not wish to incur high per-transaction marginal costs, rely on network ubiquity or performance, or require every user to buy expensive equipment. Since we cannot rely on network ubiquity, we also cannot rely on the availability of a trusted central third party to negotiate transactions, or on the availability of other peers to whom the transaction participants can appeal for opinions on trust in a decentralized manner.

In brief, Signet is built around a store-and-forward metaphor, which can process transactions in the absence of network coverage, and can reduce marginal costs by using batch transmissions. We also build a trust model based on the implied atomicity of face-to-face transactions which requires no continuous access to a trusted third party. Finally, for storage of auditable transaction traces, we use the built-in storage present on all mobile phone SIMs.

2. DESIGN

Our motivating problem can be formulated as a special case of the *optimistic fair exchange problem* [12] and more specifically of the optimistic *payment with receipt problem* [15] in which some party A wishes to exchange funds or goods for a receipt from B as seen in Figure 1, with a third party becoming involved to arbitrate between the parties when one participant misbehaves. A wants to guarantee that a receipt he receives cannot be altered or repudiated by B after the transaction is over.

The problem is modified by three facts: *a)* the transactions we are interested in occur face-to-face, providing a degree of implied atomicity, *b)* computational capacity is constrained in our context, at least on the part of one party - we may assume that a poor rural farmer cannot typically bring along a laptop or other significant computational resources to bear on a cryptological problem, and *c)* that network connectivity is not guaranteed and therefore an immediate appeal to a third-party is not necessarily possible.

2.1 Assumptions

For the purpose of our design we make several assumptions. Firstly, we assume that any participant in the system has a degree of numeracy, that is, they are able to recognize and process numbers. We feel this is a fair assumption as this is a prerequisite of making these types of transactions regardless of the receipt mechanism used.

We also assume that the central third party may be compromised by agents who have access to its keys and may collude with either party in the transaction in order to falsify information. We similarly assume that either party in the transaction may be adversarial and attempt to falsify receipts or to complete the transaction without upholding his half of the agreement.

Finally, we assume that the central third party, while potentially compromised, has a known good public key which can be relied upon to verify communications made or signed by the third party.

2.2 Components

The design of Signet uses the SIM as its centerpiece. Several considerations make the SIM ideal. SIM cards are familiar, even in the developing world where prepaid SIM vendors are commonplace, and are inexpensive. SIMs are also small and portable, and contain no moving parts or power sources which can run out. SIMs are also usable as storage media with capacities reaching 1MB [10], and can contain cryptographic coprocessors for common DES and RSA applications [10].

Most significantly, however, every SIM card used for telecommunications has its own processor, RAM, and execution environment independent of that of the host device[2], and is typically loaded once with a fixed set of applications [3], providing a secure sandbox in which applications can run without necessarily trusting the host. For the purpose of this system, a slightly more sophisticated Universal Integrated Circuit Card (UICC) which is capable of hosting multiple applications is required. UICCs are the default SIMs issued in most places today, as even most developing regions such as India [5] and Uganda [8] have 2.5G networks in major cities (supporting e.g. GPRS), which make use of UICCs. In effect, this makes the marginal cost of such a card for existing mobile phone owners zero.

The user interacts with the software on the SIM using a mobile handset, which is not necessarily a trusted component. A piece of software is resident on handsets used for the system to communicate with the SIM via APDU[3] [6].

Finally, each participant in the system receives a printed reference book containing codes corresponding to domain-specific transaction metadata. These codes are unique per user and their construction and use are detailed in the following section.

2.3 Protocol

Signet operates as follows. When parties A and B wish to conduct a receipt transaction, they meet in person. The two parties may each bring a mobile handset, or one party may provide both mobile handsets (a common case for e.g. shopkeeper banking or microfinance repayments, particularly Self-Help Groups (SHGs) where the entire group might share one handset but want to keep individual records.)

1. Each user brings his own SIM, which has been preloaded by the mobile carrier or other implementing party with the public key of the central third party as well as two uniformly random asymmetric keypairs.

2. A, the party wishing to obtain the receipt, initiates the transaction. He inserts his SIM and activates the software on the mobile handset, and enters relevant metadata $desc_g$[4] about the goods or funds to be transacted, e.g. the weight or amount of currency. $desc_g$ also contains a unique transaction ID.

[2]This is necessary at minimum to run the A3 and A8 ciphers which are part of the chain that authenticates the phone to the GSM network.

[3]APDU stands for Application Protocol Data Unit. This is the existing standard communication protocol between smartcards and host devices, including mobile handsets.

[4]N.B.: If the amounts being transacted are larger than the largest amount printed in the book, the users can simply perform multiple smaller transactions.

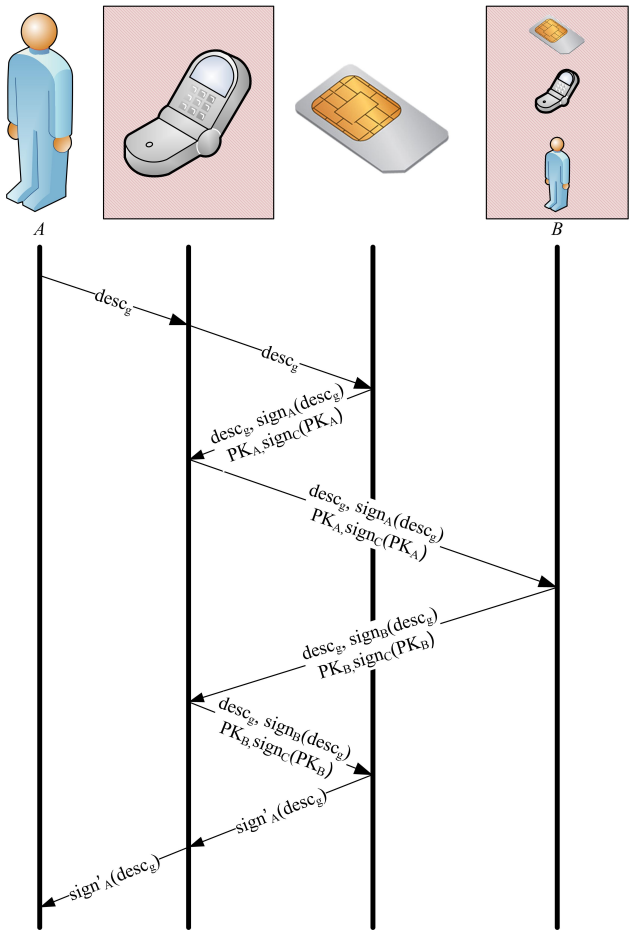

Figure 2: Simplified protocol flow. The shaded boxes indicate untrusted parties in the case of protocol execution with two phone handsets owned by one party. The analogous communication among party *B*, *B*'s SIM, and *B*'s handset are collapsed as they are all untrusted from *A*'s perspective

3. The application communicates with the SIM via APDU providing these metadata. The SIM returns this same metadata signed with the user's private key and his public key signed by the central authority. This signed key also contains a validity period to prevent users from using keys forever with no revocation mechanism.

4. A's mobile handset then sends these data to B's handset via some personal-area medium such as Bluetooth or infrared, which passes them on to B's SIM via APDU. B's SIM verifies the signature on A's public key to ensure it is valid, then checks that A's signature is properly applied.

5. B's handset then signs the metadata $desc_g$ with B's private key and appends B's public key signed by the central authority (including validity period), then sends these to A's handset.

6. A's handset sends the data to A's SIM, which verifies the validity of the keys and the authenticity of the signatures, and stores the signed $desc_g$ and B's signed

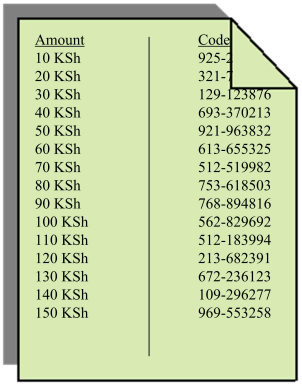

Figure 3: A rudimentary example of what a page from the reference booklet described in Section 2.2 might look like in a Kenyan context for a financial domain - microloan repayment or disbursement, banking, micropayments, etc.

key in the secure storage area of the SIM, which is accessible only by applications resident on the SIM itself.

7. A's SIM then uses a separate private key to sign the metadata $desc_g$. The last bits of this signature are then cast into an n digit number.

8. This number is passed to the handset and displayed via the application. A checks this number against the entry in his reference book corresponding to the metadata $desc_g$ originally entered, and if valid, surrenders the goods or funds he came with.

9. Either party can send his copy of the receipt to a central third party, or alternatively can hold the transaction in a queue until enough transactions have accumulated to fill an SMS message or USSD packet in order to save on marginal cost of transmission. Application-layer encryption of these SMS or USSD messages can optionally be added to protect privacy for the transaction participants. Even without this transmission however, either party can prove that the transaction took place and the conditions under which it was executed.

The value of n above is determined by the length of phone numbers in the relevant context, and displayed/chunked in the same way in order to aid in recognizing, parsing, and checking the number. For instance, if the system were to be used in Sudan, the number might be three groups of three digits each, (e.g. 811 823 492) analogous to city code and phone number as used in that context.

The rationale behind using this printed reference is to establish a trusted channel between the SIM and the user even when the handset is borrowed and therefore not trusted. A correct numeric code guarantees with high probability that the correct metadata $desc_g$ have reached the SIM and been properly recorded there. This prevents an adversary from altering software on the handset in order to arbitrarily return a simple 'OK' success code. The trust model is naturally strengthened if both parties bring their own handsets, but this in no way disrupts the protocol as it is written.

Keys are updated over the air by the carrier through standard methods; this means that any participant must periodically turn on a phone with his SIM installed in an area that has GSM coverage.

3. ANALYSIS

3.1 Security

Signet provides strong nonrepudiation of transaction receipts using standard public-key cryptography and intelligent isolation of execution environments. By using a SIM with its trusted storage and sandbox, Signet prevents or circumvents several standard attacks on signing protocols.

3.1.1 Man in the Middle

Because Signet uses a secondary signing protocol to communicate confirmation of a successfully completed transaction to the user, it is difficult for an adversary to fool the user into thinking a transaction has been successfully committed when it has not, or when the metadata have been altered, even when the adversary controls both handsets. This is critical as we wish to keep costs low by not forcing every participant in the system to have a handset; rather we expect shopkeepers, microfinance agents, etc. to carry two handsets with which to execute transactions.

Additionally, we note that as the table against which the secondary signing protocol is checked is paper-based and not electronic, it is labor-intensive to attempt to recover the second private key by checking for collisions as the table would have to be manually entered.

3.1.2 Replay Attacks

Replay attacks are difficult to execute within the Signet framework as each transaction has a unique ID within its metadata $desc_g$, which are signed. If a user attempted to insert multiple copies of the same transaction, the collision in transaction ID would cause any audit to fail. Because the metadata are signed, any edits in the transaction ID would cause the signature check to fail.

3.1.3 Privacy

It is important to note that no transaction information is sent over the air in cleartext to the GSM base station during the course of the transaction. However, we assume that any party close enough to intercept Bluetooth or infrared communication between the two handsets can ascertain the conditions of the transaction (i.e. $desc_g$) by direct observation; therefore Signet is not in the first instance meant to be a secret-keeping protocol. However, it is trivial to extend the protocol to encrypt communication with session keys negotiated by sharing signed public keys to suit this need.

3.1.4 Spoofing

A Signet transaction is difficult to spoof as they take place in person, and repeated transactions with the same party builds familiarity. The presence of a signature element means that anyone wishing to pose as another party in the system must have access to his private key in order to generate the proper signature. While it is possible for an adversary to give away his private key and collude with others to pose as him, it does not change the fact that a valid signature allows a user to verify that a transaction completed with someone in possession of a correct private key.

3.2 Economic Factors

The fact that Signet minimally requires a user to be issued a SIM card and a reference book drastically reduces the per-user cost of creating a transaction tracking system based on mobile phones. For instance in the case of an MFI with 10 agents and 100 outstanding loans, a naive system requiring each party to have a mobile phone would necessitate 110 phones, whereas Signet requires only 20.

Moreover, the fact that Signet stores transaction records and sends them only when it is the least expensive to do so reduces marginal cost per transaction dramatically in addition to allowing operation in areas where there is no mobile phone network coverage.

3.3 Auditing

In the case where transaction records stored at the central third party are lost or damaged, either party to any transaction may submit cryptographically strong proof about the transaction and its metadata, as each party has $desc_g$ signed by the other party. It is important to note, however, that this holds only so long as a public key infrastructure is in place; once no authoritative or trusted record of the central authority's public key is available, all keys signed by the central authority's private key become unverifiable. We consider this case to be a matter for disaster recovery and backup planning and therefore outside the scope of this paper.

3.4 Generalization to Fair Exchange

As the Signet protocol provides each party to the transaction a record of that transaction signed by the other party, performing general fair exchange under face-to-face atomicity reduces to including metadata about both parties' goods or funds involved in the transaction in $desc_g$. The secondary signing protocol must be adjusted to compensate for the fact that there are two quantities rather than one which need to present a unique success code, but putting codes into a 2-dimensional tabular format rather than a simple list solves most of this complexity. An alternate method is to retain the list format and formulate the transaction as a series of two linked transactions of goods for receipts.

4. IMPLEMENTATION CONSIDERATIONS

Our proof-of-concept code was implemented on a Nokia S40 5th Edition mobile handset, which supports JSR-177's [4] SATSA-APDU [1], enabling communication from J2ME to SIM-based applications via APDU. The SIM-based application was developed using Sun's Java Card Development Kit on a Java Card compliant to the 2.2.1 specification with an onboard RSA coprocessor, but in principle this could be done on any native SIM supporting multiple applications.

Given the fact that we are assuming periodic key expiration and replacement, the presence of an analog step slowing any brute-force attack on the secondary signing key and the low expected value of the given transactions, we chose to use RSA with 512 bit moduli, in deference to the onboard RSA coprocessor.

With this key length, and using $desc_g$ values with lengths of less than 256 bits, we found that the communication portion of the transaction (i.e. time spent after user interaction is complete) was approximately 3 seconds of wall-clock time, regardless of whether the handset-to-handset communication took place over infrared or Bluetooth. We consider this acceptable for the target milieu and low expected volumes.

5. RELATED WORK

Various cryptographic protocols exist [12–14, 16, 21, 27] which address the fair exchange problem in general and optimistic fair exchange, contract signing, or receipt systems in particular. Bit-by-bit approaches [16] rely on fairness based on equal computational effort of two computers trusted by their respective parties. Signet, however, allows for untrusted computational hardware other than an inexpensive and ubiquitous SIM, and the assumption of implied atomicity makes this incremental exchange unnecessary.

The optimistic approaches provide for third parties only being necessary upon failure or misbehavior, and drastically reduce message budget, which prove critical for our application. These works, however, provide slightly different guarantees and rely on different trust models (e.g. the third party is implicitly trusted and does not collude with attackers).

The Remote Transaction System (RTS) [17, 19, 20], currently deployed in Uganda, uses a smartcard-based system similar to Signet, with several key differences. Firstly, the POS hardware, held by the second party (an MFI agent) in RTS is a trusted entity, which weakens the security model. Secondly, it assumes that the third party maintaining the central server is trusted and will neither insert false transactions nor delete records of payments or goods received, even though it has the power to do either. This is a dangerous assertion in the opinion of the authors. It is also unclear what the operating costs are, as a detailed explanation of the protocol and transport layer are unavailable to the public. Finally, it is unclear what data are written to the smartcards about the transactions and whether these data are verifiable independent of the good will of the second and third parties.

Sharma et al. [25] provide a system which can verify that individual paper receipts are in fact the original pieces of paper which were issued as receipts, but do nothing to prevent standard check washing techniques [7] from being used to alter the *content* of the paper, even though the physical paper remains verifiable.

6. CONCLUSION

At present, mechanisms for proving transfer of goods in the developing world are susceptible to various forms of attack, most significant among which is forgery. In this paper we have presented Signet, a technically and cryptographically robust system which addresses this problem at low cost and without the need for continuous network coverage. Signet is deployable immediately using commodity hardware and existing technology.

We believe that Signet presents a simple and viable methodology for digital receipts while presenting users with familiar technologies and interfaces and that it presents superior security guarantees to any existing alternative. We hope to work closely with institutions in the developing world to deploy a pilot of this system in the near future.

7. ACKNOWLEDGMENTS

We would like to thank Aditya Dhananjay, Jay Chen, and Ashlesh Sharma as well as various colleagues for input and feedback in the course of preparing this paper.

We would also like to thank various personal friends and members of the microfinance community for describing the problem addressed by the proposed solution herein.

8. REFERENCES

[1] The SATSA-APDU Optional Package. http://java.sun.com/j2me/docs/satsa-dg/apdu.html, 2004.

[2] Reserve Bank of Mumbai. Department of Banking Operations and Development. Report of the Working Group on Warehouse Receipts & Commodity Futures. White Paper. http://www.rbi.org.in/upload/PublicationReport/Pdfs/62932.pdf, April 2005.

[3] Overview of the Java Card(TM) Protection Profile Collection. White Paper. http://cds-esd.sun.com/ESD24/JSCDL/java_card_pp/2.2.2-fcs/java_card_kit-2_2_2-rr-doc-protprofile-overview.pdf?AuthParam=1245009792_bf13d3d8adf3361962ef414664c0d54f&TicketId=nod2AFAYQ3R%2BluApk0GaVp6bfw%3D%3D&GroupName=CDS&FilePath=/ESD24/JSCDL/java_card_pp/2.2.2-fcs/java_card_kit-2_2_2-rr-doc-protprofile-overview.pdf&File=java_card_kit-2_2_2-rr-doc-protprofile-overview.pdf, May 2006.

[4] JSR-000177 Security and Trust Services API for J2ME. http://jcp.org/aboutJava/communityprocess/mrel/jsr177/index.html, August 2007.

[5] Airtel:GPRS Based Mobile. http://www.airtel.in/wps/wcm/connect/airtelinaes/AES/Voice+Solutions/Enterprise+VAS/Enterprise+Business+Solutions/GPRS+Based+Mobile/, 2009.

[6] APDU. http://en.wikipedia.org/wiki/APDU, 2009.

[7] Check Washing. http://en.wikipedia.org/wiki/Check_washing, 2009.

[8] MTN Uganda website: A bright Yello World. http://www.mtn.co.ug/MTN-Products/MTNinternet/Personal/MTN-Mobile-Internet.aspx, 2009.

[9] Safaricom: M-PESA. http://www.safaricom.co.ke/index.php?id=745, 2009.

[10] Samsung Smart Card IC. http://www.samsung.com/global/business/semiconductor/support/brochures/downloads/systemlsi/smartcardic_061107.pdf, 2009.

[11] Village Phone: Connecting Technology and Innovation. http://www.grameenfoundation.org/what_we_do/technology_programs/village_phone, 2009.

[12] N. Asokan, M. Schunter, and M. Waidner. Optimistic protocols for fair exchange. In *CCS '97*, pages 7–17, Zurich, Switzerland, 1997. ACM.

[13] N. Asokan, V. Shoup, and M. Waidner. Asynchronous Protocols for Optimistic Fair Exchange. In *Security and Privacy '98*, pages 86–99, Oakland, CA, 1998. IEEE.

[14] G. Ateniese. Efficient Verifiable Encryption (and Fair Exchange) of Digital Signatures. In *CCS '99*, pages 138–146, Singapore, 1999. ACM.

[15] H. Burk and A. Pfitzmann. Value exchange systems enabling security and unobservability. *Computers and Security*, 9(9):715–721, December 1990.

[16] S. Even and A. Goldreich, O. ad Lempel. A Randomized Protocol for Signing Contracts. *Communications of the ACM*, 28(6):637–647, June 1985.

[17] L. Hewlett-Packard Development Company. Remote Transaction System Solution Brief. White Paper. http://www.sevaksolutions.org/docs/RTS%20HP%20Solution%20Brief.pdf, 2005.

[18] C. F. Kaestle. The History of Literacy and the History of Readers. *Review of Research in Education*, 12(1):11–53, January 1985.

[19] M. Kam and T. Tran. Lessons from Deploying the Remote Transaction System with Three Microfinance Institutions in Uganda. In *Proceedings of UNIDO-UC Berkeley "Bridging the Divide" Conference, 2005"*, Berkeley, CA, 2005.

[20] N. Magnette and D. Lock. What Works: Scaling Microfinance With the Remote Transaction System. http://www.digitaldividend.org/pdf/rts.pdf, August 2005.

[21] O. Markowitch and S. Saeednia. Optimistic Fair Exchange with Transparent Signature Recovery. In *Financial Cryptography*, volume 2339/2002, pages 339–350. Springer Berlin / Heidelberg, 2002.

[22] Murray, John E. Generation(s) of Human Capital: Literacy in American Families, 1830-1875. *Journal of Interdisciplinary History*, 27(3):413–435, Winter 1997.

[23] T. S. Parikh. Rural microfinance service delivery: Gaps, inefficiencies and emerging solutions. In *ICTD '06*, pages 223–232, Bangalore, India, 2006. IEEE.

[24] Rosin, R. Thomas. Gold Medallions: The Arithmetic Calculations of an Illiterate. *Council on Anthropology and Education Newsletter*, 4(2):1–9, Jul. 1973.

[25] A. Sharma, L. Subramanian, and E. Brewer. Secure Rural Supply Chain Management Using Low Cost Paper Watermarking. In *NSDR '08*, pages 19–24, Seattle, Washington, 2008. ACM.

[26] D. Talbot. Upwardly Mobile. *Technology Review*, 111(6):48–54, November/December 2008.

[27] D. Yum and P. Lee. Efficient Fair Exchange from Identity-Based Signature. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, E91-A(1):119–126, 2008.