Usably Secure, Low-Cost Authentication for Mobile Banking

Saurabh Panjwani, Edward Cutrell
Microsoft Research India
{saurap,cutrell}@microsoft.com

ABSTRACT
This paper explores user authentication schemes for banking systems implemented over mobile phone networks in the developing world. We analyze an authentication scheme currently deployed by an Indian mobile banking service provider which uses a combination of PINs and printed codebooks for authenticating users. As a first step, we report security weaknesses in that scheme and show that it is susceptible to easy and efficient PIN recovery attacks. We then propose a new scheme which offers better secrecy of PINs, while still maintaining the simplicity and scalability advantages of the original scheme. Finally, we investigate the usability of the two schemes with a sample of 34 current and potential customers of the banking system. Our findings suggest that the new scheme is more efficient, less susceptible to human error and better preferred by the target consumers.

Categories and Subject Descriptors
D.4.6. [**Security and Protection**]: Authentication, Cryptographic controls. H.5.2 [**User Interfaces**]: Evaluation/methodology, Input devices and strategies. J.7 [**Computers in Other Systems**]: Consumer Products. K.4.4 [**Electronic Commerce**]: Cyber-cash, digital cash, Security.

General Terms
Algorithms, Design, Security, Human Factors

Keywords
Mobile, banking, authentication, PIN, paper, security, usability, developing regions, ICTD.

1. INTRODUCTION
In the developing regions of the world, there are over a billion people who do not hold a bank account but who still own and use a mobile phone on a regular basis [1]. Increasingly, we are seeing new ventures which make modest banking facilities available to such people by utilizing mobile phones as the primary instrument for conducting transactions. Through a network of human agents who facilitate cash deposits and withdrawals, these systems extend the reach of banks to remote areas in a manner that is not only more convenient for consumers, but often less expensive than conventional methods [2].

Symposium On Usable Privacy and Security (SOUPS) 2010, July 14-16, 2010, Redmond, WA, USA.

While the idea of using mobile phones as banking instruments is quite fascinating and has multiple potential benefits, there are some unique challenges encountered when it is implemented on the ground. One key problem is fraud prevention. At the very least, every bank that provides a mobile banking service would like to ensure that transaction requests generated from a phone are made by the legitimate owner of the phone, and not by some ill-meaning impostor. In fact, in a few countries, there are strict government stipulations for user authentication which banks must meet in order to provide mobile banking services [3].

In conventional ATM-based banking, the primary tool used to authenticate users over a network is the personal identification number (PIN) – a 4-digit secret password which every user is required to remember and communicate to the bank before conducting any transaction. Along with the PIN, the user must establish possession of another identification token (e.g., the ATM card issued by the bank), and these two entities together form a simple 2-factor authentication mechanism for enabling transactions. Although there have been numerous efforts to replace the use of PINs with other techniques (e.g., user biometrics [4],[5]), such efforts are far from gaining universal adoption. Owing to its simplicity and economy of implementation, the PIN today remains the de facto standard as a user authentication tool in banks across the world.

Naturally, PINs are a candidate tool for authenticating users in mobile banking as well and in fact, most existing mobile banking services use them, too [6-9]. However, to guarantee secrecy of PINs, they need to be suitably protected when transmitted by banking applications over the mobile network. The challenge lies in designing PIN-protection schemes which can function on *any* phone that potential customers may own and which offer an interface that is usable by populations with low literacy. There are several roadblocks to accomplish this. First, phones' in-built encryption services cannot be relied upon since these services provide security only at the network layer (as opposed to application-level security) and even where they do, the security offerings are not robust [10-12]. Second, a large fraction of phones in the developing world have limited computing and storage capabilities, and so, encryption methods used by ATMs are not easy to deploy on them. And finally, even if banks opt for the simplest software-based solutions for protecting PINs, it is practically impossible for them to program the phone of every customer they cater to unless they garner support from network operators for it. Relying on operator support for programming phones, in turn, has its share of limitations[1].

[1] Relying on operator support (e.g., for programming SIM modules of phones) does not yield an end-to-end secure solution since operators

In India, a mobile banking service provider named EKO is using a unique and cost-effective solution for authenticating users over GSM-based phone networks [6]. EKO is a business correspondent of State Bank of India (SBI), the leading public sector bank in India, and through its mobile banking system, it currently services over 65,000 customers with a daily transaction volume of nearly 1,500,000 INR (33,000 USD). EKO's banking system is operator-independent (it works across multiple mobile operators) and phone-agnostic (requiring only GSM-capability from phones) and it is currently the largest such mobile-based banking service in India. The service relies on PINs for authentication, and uses a unique paper-assisted method for transmitting PINs over the network: each user holds a unique paper codebook containing random 6-digit one-time passwords – henceforth, referred to as *nonces*[2] – and every time he needs to authenticate himself, he transmits a 10-digit number formed by juxtaposing his PIN with a fresh nonce from the codebook. (See figure 1.) EKO's authentication system is endorsed by Verisign and by the end of 2010, it may be used by 100,000 customers across northern and eastern India.

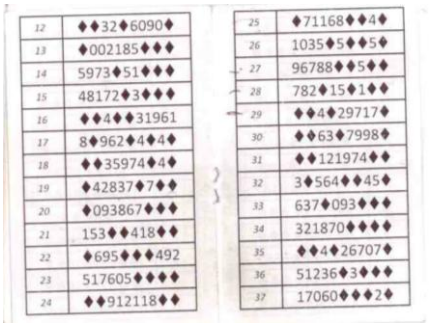

Figure 1. An illustration of the scheme EKO uses to authenticate users to banks. The scheme requires users to juxtapose PINs randomly with 6-digit nonces stored in a paper codebook, each user holding a unique codebook. For example, if the user is currently using the 13th nonce in the codebook (marked ♦002185♦♦♦), and his PIN is 6391, his signature for the current transaction would be 6002185391.

1.1 Our Contributions

In this paper, we report a security weakness in the authentication scheme used by EKO and show that it is susceptible to easy and efficient PIN recovery attacks. We demonstrate an algorithm which enables EKO's banking agents as well as network eavesdroppers to recover a user's PIN by observing at most 7 transaction messages created by that user. PIN recoverability severely reduces the strength of the authentication scheme and could result in impersonation attacks if users' codebooks are compromised.

Next, we propose a new authentication scheme which also relies on the use of PINs and printed codebooks but which provides better protection to PINs as they are transmitted over the network. Our solution is a variant of the one-time pad, a classic encryption scheme which is known to provide unconditional security, and it requires users to perform simple substitution-based coding of the PIN before entering it into the phone. (See figure 4.) Although the one-time pad is rather storage-inefficient for general-purpose encryption, in the context of PIN protection, we find that its efficiency is quite reasonable and in particular, is comparable with that of EKO's current PIN-juxtaposition method. Using codebooks for encrypting PINs in this manner results in a secure 2-factor authentication scheme: users must possess the right phone and the right codebook *and* know the right PIN in order to authenticate themselves.

Our authentication scheme protects PINs not only from eavesdroppers on the network (including, in particular, the network operator) but also from banking agents who often mediate transactions between users and the bank and are thus a critical source of threat to mobile banking systems. In addition to protecting PINs from agents and network operators, the scheme enjoys other security properties like resistance to skimming attacks and shoulder-surfing attacks, which makes it of potential interest in domains outside of mobile banking. In particular, our scheme offers a more secure alternative to the authentication solutions used by current-day ATM machines, which are known to be susceptible to such attacks.

Both the schemes we consider – EKO's current authentication scheme and the one that we propose – require users to modify their PIN using a printed nonce and to enter the modified PIN on a mobile phone before a transaction can be successfully conducted. This modification task could potentially pose a usability barrier for deployment. In order to understand and address this concern, we conducted a usability study with 34 current and potential mobile banking customers in the regions where EKO currently operates. Our goal in this study was to gauge users' perception of the two systems and to measure the efficiency and accuracy with which they are able to operate them. The results from our study suggest that our authentication scheme outperforms EKO's scheme both in terms of efficiency of PIN entry and in terms of the rate of human errors. Not only this, 65% of our users expressed a preference for using the new method over the existing one. Our scheme thus provides an alternative to EKO's current system which not only offers better PIN-privacy guarantees but is also more usable by the target consumers.

The outcomes of our security investigation and the details of the new scheme have been communicated to EKO. Indeed, the usability study reported in this paper was conducted in collaboration with EKO and a design and research firm in India named CKS India Pvt. Ltd. Since the new scheme has been found to enjoy several advantages over the existing one, it is likely that this scheme will be deployed by EKO in the near future.

2. BACKGROUND AND RELATED WORK

The idea of using mobile phones for banking has been in practice for almost a decade in the developed parts of the world [13]. While the phone is used in the developed world to make banking *convenient* for people who already hold bank accounts, in the developing world it has found a new application – that of

may have access to information transmitted over the network. Besides, it entails the challenge of ensuring inter-operability between phones under different operators or else standardizing a solution that works across multiple operators. Both these options are difficult to pursue, given the competitive landscape in the mobile services market in the developing world.

[2] A nonce refers to a random **n**umber that is to be used only **once** in any cryptographic scheme.

providing *access* to banking services for those who cannot bank using traditional methods. Initially popularized by the M-PESA service in Kenya [7], the concept has now spread to several countries across Asia and Africa including South Africa [8], the Congo Republic [14], the Philippines [9], Pakistan [15] and India [6]. Mobile banking has now become a significant conduit for monetary flows in the countries where it operates: M-PESA alone is reportedly mobilizing 10% of Kenya's GDP through its banking network [16] while G-Cash in the Philippines is transacting 100 million USD worth of money on a *daily* basis [17].

All these systems provide the key services available in regular banking including the facility to deposit and withdraw money and the facility to transfer money across accounts. Furthermore, there are no lower limits on account balances and the procedure for account creation is often simplified, improving access for low-income populations. In order to prevent misuse of the service, an upper limit on balances as well as on transaction volumes is typically imposed, but this limit varies from country to country.

Figure 2. Mobile banking in India: (left) a mobile services shop whose owner is also an agent for EKO; (right) a deposit transaction being conducted by an agent (who runs a stationary shop) as the depositing customer waits for the bank server's acknowledgement via SMS. *(Credits: CKS India Pvt. Ltd.)*

Every account is identified by the corresponding user's mobile phone number and the balance in a user's account is equivalent to the amount of *stored value* maintained against his/her phone number (much like the way stored value is maintained in prepaid cards). The conversion between physical cash and stored value happens with the help of designated *agents* who typically are also account holders and who use their accounts to exchange real cash for stored value as desired. For example, a user wishing to deposit amount x into his account would approach a nearby agent, submit the respective amount in cash and have the agent transfer a stored value of x from his account into the user's account through a suitable transaction message sent on the mobile network. This transaction message is sent from the *agent's* phone to a bank server on the network. Later, if the user wishes to withdraw amount y (for some $y < x$) from his account he would approach the same or another agent, transfer a stored value of y from his account into the agent's account and receive the cash equivalent in return. In this case, the transaction message would be sent from the *user's* phone to the bank server. Agents thus function like the human analogues of ATM machines – authorizing deposit transactions, and executing withdrawals, when suitably authorized by the user. Besides facilitating deposits and withdrawals, agents are also responsible for initiating new users into the system and implementing the requisite background checks.

In all mobile banking systems we are aware of, the communication between the users' phones and the bank server is implemented using GSM-based services like SMS or USSD[3] and acknowledgements from the bank server are also sent using the same channels. In order to protect the system from forgery, banks must implement methods to ensure that the sender of every transaction request SMS or USSD message can be accurately verified. Different mobile banking systems use different authentication solutions depending upon the amount of control they have on the network protocols and the regulatory climate in their region. The pioneer in this space, M-PESA, uses a PIN-based approach to authenticate users to the bank: messages are sent using USSD and since the provider of the service, Safaricom, has complete control of the network, a proprietary encryption program is installed on users' SIM cards to protect the PIN during transmission. (Details of the encryption are not publicly known.) In the case of G-Cash in the Philippines [9], PINs are used, too, but they are transmitted in plain as part of SMS-based transaction messages. Such a solution does not guarantee good security since GSM's inbuilt encryption algorithms have several reported weaknesses [10],[11], which may cause PINs to be compromised. In India, network operators are banned from offering mobile banking solutions (only banks can do so) and on top of that, stringent norms for security practice are imposed by law. This naturally rules out both the M-PESA and G-Cash paradigms of security and has led to mobile banking companies (like EKO) to design their own application-level solutions for authentication. The extent to which such solutions undergo security audit by regulating authorities is neither well understood nor publicly documented.

The use of nonce-based tokens for remote authentication, as done by EKO, is an established cryptographic technique, used in several corporate access control systems. The most popular amongst these is the RSA SecurID [18], wherein users are provided a tamper-proof electronic dongle with a small LCD display that displays a 6-digit dynamic nonce. For authenticating themselves to a remote server, users are required to provide their password along with the current nonce displayed by the dongle, and both the values are well-protected (typically through VPNs) while being transported to the server. The nonces are generated by the dongle using a proprietary algorithm that utilizes an inbuilt clock and a fixed pseudorandom seed. In contrast, the nonces used in EKO's system are all pre-generated in bulk by Verisign and distributed in the form of paper codebooks.

There have been some proposals to use voice biometrics for authenticating users in mobile banking [19],[20]. However, to the best of our knowledge, none have been deployed at scale. Even though voice-based authentication systems have been shown to work in the laboratory [21], the problem of ambient noise in developing world environments makes them extremely difficult to deploy in mobile banking. Some companies in India currently use fingerprint biometrics to authenticate users in agent-assisted banking [22],[23], but the setup and operational costs of these solutions are significantly greater than that of token-based systems

[3] USSD stands for Unstructured Supplementary Service Data, a communication service provided on GSM phones for some operator-defined supplementary services (e.g. carrying out balance checks for prepaid phone cards). USSD is faster than SMS, and free of cost.

and these solutions are not implementable over low-end mobile phones, which are prevalent in the developing world.

3. THE AUTHENTICATION SCHEMES

In this section, we describe EKO's current authentication scheme, report weaknesses in it, and present a new scheme, which provides better security of PINs than the current one.

3.1 EKO's authentication scheme

Security in EKO's mobile banking system hinges on two authentication tools – a numeric 4-digit PIN and a printed codebook containing a list of 6-digit nonces, as shown in figure 1. Each user has a unique PIN and a unique codebook assigned to him securely through out-of-band communication. To explain how these tools are used, we first provide some basics about EKO transactions. Recall that in any mobile banking system, a transaction message sent by a user is meant to request the bank that a certain amount of stored value be transferred from his/her account into another user's account. (For withdrawal transactions, users send the transaction message, while for deposit transactions, agents send the transaction message.) In EKO's case, every transaction message is sent using the USSD service and contains 4 principal fields – a 3-digit USSD code identifying the intended route of the message; a 10-digit phone number identifying the account to which stored value is to be transferred; the amount to be transferred; and finally, a *signature*, which is meant to authenticate the user to the bank. The fields are all numeric and are separated by *'s and terminated by a #. In all, every transaction message contains 29 to 32 characters.

The signature is a 10-digit number which is freshly created per transaction as follows:

Signature formation in EKO's scheme:

1. Look up the first unused nonce in the codebook. It has 4 diamond-shaped blanks juxtaposed with it.
2. Recall your PIN and replace the diamond-shaped blanks with the 4 digits of the PIN in order. The resulting 10-digit number is the signature for the current transaction.

For example, if the user is currently using the 13th nonce in the codebook shown in figure 1 (marked ♦002185♦♦♦), and his PIN is 6391, his signature would be **6**002185**391.**

Nonces are generated using a pseudorandom number generator at a server operated by Verisign and 50 such nonces are printed in each codebook. Each codebook has a unique ID and users are provided a fresh codebook at the time of registration and every time a codebook gets exhausted. Upon receipt of a new codebook, users send a registration message to the bank server (over USSD); this message contains the ID of the codebook and a signature formed using the *first* nonce in the codebook. The registration message helps the bank form an association between a user and his/her current codebook and also helps record the user PIN at the time of registration.

EKO agents guide users to select strong, hard-to-guess PINs although to what extent this happens is not well-documented. In the future, the plan is to have PINs assigned by the bank, as is the case in ATM-based banking.

Since most of EKO's consumers have limited literacy and thus face difficulty in forming and sending long transaction messages – as is required for withdrawals – EKO implements a facility for *aided* transactions. These are withdrawal transactions wherein the transaction message is sent from the *agent's* phone but the message is preceded by a special code to distinguish it from a regular message. In aided transaction, agents specify the phone number of the user who wishes to withdraw money and the signature is obtained from the same user via *oral* communication; thus the bank still views the transaction request as being authorized by the user. EKO reports that since the launch of this facility, 67% of withdrawal transactions have been conducted in an aided manner.

3.2 The security flaw

EKO's current authentication scheme is susceptible to easy PIN-recovery attacks. We find that given access to at most 7 withdrawal messages made by a single user, the PIN of that user can be recovered with probability nearly 1.

The attack is fairly straightforward: Given a list of k 10-digit signatures corresponding to a user, the attacker exhaustively searches for 4-digit subsequences that are common to all the signatures. If he finds that there is only one such subsequence, it returns it as the user's PIN. Else, he waits for the $(k+1)^{th}$ signature and repeats the procedure.

Our finding is that even for relatively small values of k (namely, $k=7$), the success probability of the attack is very high. In a laboratory simulation of EKO's scheme, we generated codebooks and PINs at random (using C#'s inbuilt pseudorandom generator) and implemented our attack on 10,000 sets of codebook-PIN combinations. We found that in 99.67% cases, the PIN was recoverable given just the first 7 signatures formed using the codebook. In practice, the value of k required to recover the PIN may be even smaller than 7: the interspersion of the PIN with the nonce may not be perfectly random or the nonces themselves may be lacking in entropy. In an independent investigation of real EKO codebooks [24], we found that on average, PINs can be recovered with 100% certainty given roughly 4 transaction signatures only, and this is true for every value of the PIN.

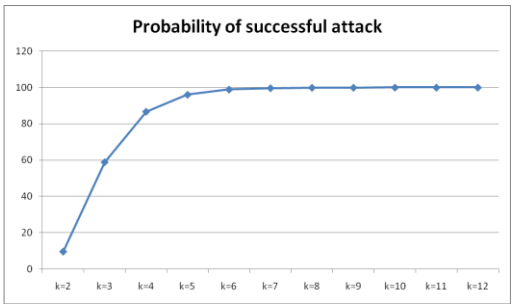

Figure 3. Success rates of the PIN-recovery attack (represented as percentages), as computed in a lab experiment.

There are two possible ways in which the attack can be mounted in practice. For one, agents can recover PINs of users who use the aided transaction facility: since users tend to transact through the same agent repeatedly, agents can acquire multiple transaction signatures for all users they service, and use this information to recover their PINs. Besides agents, arbitrary eavesdroppers on the phone network can store messages sent over time (through the use

of appropriate hardware) and after acquiring enough messages from a single user, they could recover that user's PIN.

Our conclusion is that the PIN-juxtaposition technique used by EKO does not provide significant protection to PINs; in fact, the security provided is only marginally better than a scheme in which PINs are not used at all! We remark that the compromise of a user's PIN does not imply easy impersonation as that user; doing so still requires access to the user's codebook and his phone. However, the fact that PINs *can* be compromised in EKO's scheme does weaken its claim for 2-factor security considerably. We now present a scheme which addresses this concern.

3.3 Our proposal

In our scheme, just like in EKO's, each user holds a codebook containing a list of nonces though in this case each nonce in the codebook is a *10-digit* number. To facilitate authentication, we store the nonces in a manner that enables users to quickly look up digits at arbitrary positions within the nonces. One possible approach, illustrated in figure 4, is to place the digits 0,1,2,…,9 right above the digits of every nonce; these digits serve as position numbers for the digits in the nonce. So, for example, in the 21st nonce in the figure, 3527850631, the digit at the 0th position is 3, that at the 1st position is 5 and that at the 6th position is 0. Each user also holds a secret 4-digit PIN, as before. For authenticating himself to the bank, the user creates a numeric signature combining the PIN with the codebook content as follows:

Signature formation in the new scheme:

1. Look up the first unused nonce in the codebook.
2. Recall your PIN, say $x_1x_2x_3x_4$, and return the 4 digits which are located in the x_1^{th}, x_2^{th}, x_3^{th} and x_4^{th} positions of the nonce, in that order. The resulting 4-digit number is the signature for the current transaction.

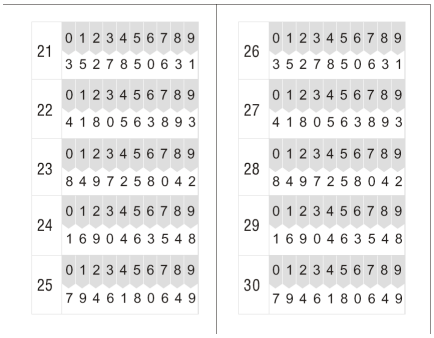

Figure 4. An illustration of the proposed authentication scheme. The scheme involves encrypting PINs using 10-digit nonces. For example, if the user's PIN is 1230 and the current nonce is the 21st one i.e. 3527850631, the encrypted PIN would be the result of looking up the 1st, 2nd, 3rd and 0th positions in the nonce, namely 5273.

The bank stores PINs and codebooks of all users and when it receives a transaction message from a particular user, it repeats the signature computation operation (using the first unused nonce in the user's codebook) and checks if its output is the same as the signature in the incoming message. Authentication succeeds if and only if this is the case.

The transformation from PINs to signatures in our scheme is meant to conceal information about the PIN and to protect it from recovery during transit. The transformation function itself is a variant of the substitution cipher, which is one of the oldest cryptographic schemes devised by man and has been used as an encryption tool even before computers were invented. (Julius Caesar used a type of substitution cipher more than 2000 years ago!) There are, however, some notable differences between our scheme and the substitution cipher. First, we operate on numeric data only, which makes the substitution task easier to perform, and more efficient in terms of key storage. Second, the transformation function need not be a permutation which means that it is not strictly a cipher (unique deciphering is not possible). Restricting the function to be a permutation – that is, requiring each digit of a nonce to be distinct from the others – has both pros and cons. On the positive side, it enables the implementation of a deciphering facility at the bank server which means PINs need not be stored in plain, but could be stored in a cryptographically-hashed manner. However, on the negative side, such a restriction leads to poorer security guarantees against impersonation attacks. We chose to eliminate this restriction primarily for simplicity of implementation.

Finally, one key difference between our PIN transformation system and the substitution cipher is that we use a new transformation function for each transaction, whereas in the latter, the same function is used repeatedly. Since the nonces are random and independent, this is effectively the same as applying a one-time pad on the PIN and thus makes the scheme more robust against impersonation attacks than using a fixed substitution key. (Applying a fixed key on the PIN for each transaction would make it easy for eavesdroppers to impersonate arbitrary users.) We note that the traditional one-time pad scheme – in which nonces are modularly added to the plaintext – would require shorter nonces and would, in fact, be more storage-efficient than the one we propose. However, such a scheme is likely to encounter severe usability challenges since it requires the ability to perform modular addition mentally, a skill that is arguably difficult to teach to users with limited education backgrounds. On the contrary, substitution coding is a skill that is known to have been used by humans in practical systems for a long time, although its usage in the above form, and with a developing-world population, remains untested prior to the current work.

Note that signatures created in our authentication scheme are only 4 digits long, as opposed to the 10-digit EKO signatures; we expect that this feature will make our scheme more usable than EKO's scheme in practice. We now expand upon some key implementation issues, and discuss the security properties of our scheme.

3.3.1 Digit distinctness

Our authentication solution provides differential security based on the number of repeated digits in the user's PIN. In particular, PINs which have repeated digits (like 1111) map to signatures with repeated digits (like 2222 or 3333), which make both the PINs and the signatures easier to guess. For security reasons, we thus recommend that the system be deployed with a mandate that all PINs have distinct digits. There are 5040 such PINs, a space that is large enough to counter dictionary attacks.

3.3.2 Synchronization issues

As in EKO's scheme, the bank server needs to be synchronized with the user for accurate authentication. To accomplish this, each nonce is labeled with a unique sequence number (as in figure 4) and users must use nonces in order of their sequence numbers. The sequence numbers are maintained at the bank's end as well. If a user goes out of sequence, the bank sends an error message via SMS and this message contains the sequence number of the nonce being expected. To prevent dictionary attacks by unauthorized users, a standard locking mechanism is implemented – the bank accepts at most 3 consecutive incorrect signatures, after which the account is deactivated and can be re-activated only through out-of-band communication.

3.3.3 Nonce deletion

For best security guarantees, every nonce must be deleted right after it has been consumed for successful authentication. This can be achieved using various possible designs. One possibility is to store them in a booklet with perforated sheets – the perforations would enable users to discard the nonces that have been used for authentication. Another possibility is to use scratch-off cards or stickers, one sticker per nonce; the sticker would need to be peeled or scratched off immediately after use. Yet another possibility is to store all nonces in a paper roll, encased in a solid body with a small window; the window would permit the user to view the nonce at the edge of the paper roll (the "current" nonce) and subsequently tear off the portion of the roll which contains that nonce. (This would be a miniature form of receipt dispensers in point-of-sale devices.) A final possibility is to use electronic hardware to store and display nonces, much like the dongles that are used in electronic tokens like RSA SecurID [12]; in this case, nonce deletion could be accomplished using software.[4]

Implementing a nonce deletion facility, besides improving security of the scheme, helps alleviate the issue of synchronization by enabling users to keep track of the first unused nonce. We remark that faithful deletion of nonces does not improve security in EKO's scheme in a significant way, whereas in our scheme, it is quite beneficial.

3.3.4 Security

A complete analysis of the security of our scheme is presented in a separate publication [24]. Here, we report the key outcomes of the analysis. First, our scheme provides much better security against PIN recovery attacks than EKO's scheme – the success probability of the best attack here is roughly 10^{-4} whereas in the case of EKO's scheme, there exists an attack with success probability nearly 1. We also analyzed our scheme's security with respect to impersonation attacks, wherein a malicious user acquires a user's phone and/or codebook and tries to use this information to authenticate as the user to the bank. If the attacker manages to acquire both the phone *and* the codebook, the chances of impersonating the corresponding user are much greater in EKO's scheme – a probability of nearly 1 against a probability of roughly $10^{-3.7}$ in our scheme. (This is because EKO's scheme provides poor protection of users' PINs, whereas our scheme does protect them.) If the attacker acquires only the user's phone and not the codebook, then both schemes are secure against impersonation, although the success probability of an attack is greater in our scheme than in EKO's: 10^{-4} versus roughly 10^{-8}. This is not a matter of grave concern since with a suitable account locking mechanism in place, an attack probability of 10^{-4} suffices for most applications; for example, it is the standard in ATM-based banking. Variants of our scheme which provide even better impersonation-resistance are proposed in [24] although these variants are likely to be less usable than the current scheme.

Our solution provides some notable security benefits over traditional PIN-based authentication methods, such as those used in ATM-based banking. Since it requires every user to encrypt his PIN prior to entering it into the system, the chances that PINs can be leaked by tampering with access terminals are substantially reduced. In particular, the solution is secure against skimming attacks wherein counterfeited access terminals are installed in place of genuine ones and used to capture secret information of users for later impersonation to the bank. Skimming attacks are currently the most dominant cause of fraud in ATM-based banking and in 2009, they accounted for a loss of more than one billion dollars to banks worldwide [25].

The other benefit of having users encrypt their PINs themselves is that PINs are less likely to be stolen by mere observation of the PIN-entry process via shoulder-surfing. Shoulder-surfing attacks are another real threat in banking transactions and there is a rich literature on techniques to counter these attacks (see, for example, [26-28]). However, most of these techniques make changes to the hardware or the software (or both) of the access terminal used for PIN-entry whereas our solution offers an alternative based on the use of supplementary tokens. The downside of a token-based PIN-encryption protocol, is that it comes at the cost of reduced usability. However, this cost may be bearable in cases where the information to be protected is small (like a 4-digit number) but the potential damage caused by its compromise is immense.

We remark that the security analysis of our scheme presented in [24] is with respect to an attack model wherein adversaries can eavesdrop on the communication channel and acquire users' phones and/or codebooks. However, as noted in [24], one could consider more severe attacker capabilities like caller ID spoofing and real-time interception and modification of messages (man-in-the-middle attacks). Such possibilities, though interesting from a theoretical perspective, are difficult to mount in current-day mobile networks. Our authentication scheme does not guarantee strong security against adversaries who can spoof caller IDs or mount man-in-the-middle attacks. We believe that designing mobile banking systems which are secure against such attackers but which do not involve software installation on phones is non-trivial and we leave it as an open problem.

4. STUDY METHODOLOGY

In comparison to typical PIN-based authentication solutions – wherein a user simply types in his PIN onto an access terminal – the two schemes we have been considering here require the user to do additional work. A natural question to ask is: which of the two schemes provides a better interface to the user, and how do these interfaces compare with the traditional PIN-entry interface?

[4] While electronic forms of nonce storage may appear appealing, the current rate of utilization of nonces in EKO transactions does not seem to be high enough to make them more cost-effective than paper tokens. This is because nonces are used by customers only to do withdrawals and money transfers, which are relatively infrequent, compared to deposit transactions.

We conducted a user study with current and potential mobile banking customers in India to answer this question. Our study ran over a period of two weeks and was conducted in two different locations in northern India. This section provides details on the study design, including our user sample, our task definitions, and the hypotheses we tested.

4.1 Sample

We recruited 34 participants from 2 different regions in India – Uttamnagar (Delhi) and Sitamarhi (Bihar) – where EKO is currently operating. Fifteen participants were sampled from the Delhi region and nineteen from Bihar. Participants came from three different categories:

- ***Agents:*** There were 8 participants (5 from Delhi, 3 from Bihar) who had been EKO agents for at least one month at the time of the study. These were sampled with a human bias which ensured that the current demographic spread of agents is well represented – 5 of them run mobile services and retail shops, 2 run stationary shops and one runs a pharmacy. Their ages range from 23 to 45, with the mean age being 32.4. Monthly incomes range from 6,000 INR (120 USD) to 25,000 INR (500 USD).
- ***Existing customers:*** There were 13 participants (5 from Delhi, 8 from Bihar) in our sample who had been EKO customers – but not agents – for at least one month. These participants were selected randomly from EKO's customer list and they come from a variety of backgrounds and included marketing agents, students, shop owners, an electrician, a sweeper, a news reporter and one housewife. Their ages range from 18 to 38, the mean age being 27.4. Their monthly income is more modest – a range of 1,000 INR (20 USD) to 10,000 INR (200 USD).
- ***Potential customers:*** Finally, we picked 13 participants (5 from Delhi, 8 from Bihar) who were candidates for becoming EKO customers in the near future. These were sampled with the help of EKO agents based on reported interest in EKO's service. (The eight agents in our sample provided us a list of candidates each, and we sampled randomly from this list.) Potential customers have a profile similar to that of existing customers: sales agents, shop owners, students, vegetable vendors, a sweeper, a laborer, a cook in a restaurant and a housewife. Ages range from 20 to 36, the mean being 27.4. Monthly incomes range from 1,000 INR (20 USD) to 7,500 INR (150 USD).

Our subjects had very little formal education and at least 12 participants reportedly did not go to school beyond 10^{th} grade; eight of these completed only primary schooling. The agents were more educated, all but two of them having been through college. Participants had very limited fluency in English, so all verbal interactions were conducted in Hindi, the national language of India, with which participants were fluent. There were 32 males in our sample, and 2 females; this is closely representative of the demographics of current EKO consumers, of which all agents are males, and among non-agents only 15% are females in Delhi, and 7% in Bihar. One of the females was an existing customer, and one a potential customer.

4.2 Task Definitions

We conducted a within-subjects comparison of participants' performance on three types of tasks: plain PIN entry, signature formation using EKO's scheme, and signature formation using our scheme. Our primary goal was to measure the time they take to perform these tasks and the rate at which they make errors in each of the tasks.

Ahead of all tasks, participants were shown 2 PINs – a *simple* PIN (0123) and a *complex* PIN (6183) – which they were asked to memorize and later use to perform the tasks. We used 2 identical phones across all participants – one for Delhi, one for Bihar. The tasks were as follows:

1. ***Plain PIN entry***: In this task, participants typed in the PIN they had memorized into the mobile phone. Participants first performed the task with the simple PIN (three times) and then with the complex PIN (another three times). We noted the time taken for each of the task trials using a digital stopwatch. We provided a simple cue to begin task execution and recorded task completion based on physical observation.
2. ***EKO signature formation***: Here, participants were presented with a codebook of the sort currently being used by EKO (figure 1) and there were 2 identical codebooks we used across all participants – one for Delhi, one for Bihar. Participants were first trained to form signatures as done in EKO's authentication scheme and enter it into the phone. Because existing customers were ostensibly already familiar with the current EKO authentication scheme, we used two fixed training protocols – one for current customers and one for potential customers. Each protocol involved demonstration of task execution by the researcher multiple times: twice for existing customers, 3 times for potential customers. After being trained in this manner, participants formed multiple signatures using consecutive nonces from the codebook, starting with the same nonce and proceeding in sequence.

   Participants often made errors (i.e., they entered the wrong signature) and the experimenter pointed out errors after a task trial was over (not during the task trial). The task was performed until participants could conduct three *consecutive* error-free trials, after which we assumed that they were adequately trained. After doing the task with a simple PIN, they repeated the task using the complex PIN, again up to a point of three consecutive correct entries. All task trials were timed as in the case of plain PIN entry.
3. ***New signature formation***: In this task, participants were provided a codebook for the proposed authentication scheme, of the type shown in figure 4. Participants were first trained to perform the task and *all* participants received identical training – 3 demonstrations of task execution. After being trained, participants performed the task on their own multiple times, using consecutive nonces from the codebook, while we timed them. In case of erroneous entries, the experimenter pointed out mistakes after signature entry was complete. The task was performed till the point of 3 consecutive error-free signature entries. As above, the complex PIN version of the task followed the simple PIN version.

Care was taken to order the tasks suitably: Plain PIN entry – the baseline task – was always performed first by each participant, but

the other 2 tasks were counter-balanced to eliminate ordering effects. Besides noting errors made by participants, we also recorded all *edits* performed by them during signature entry: an edit comprised an event in which the user used the backspace button on the mobile phone keypad to change some previously-entered digits. We also had participants perform two *transaction tasks* besides the tasks listed above. In one, participants typed out a complete transaction message for a withdrawal operation – in the format defined by EKO – and they authenticated themselves using EKO's authentication scheme. In the other, they typed out a similar transaction messages (for the same withdrawal amount) but they authenticated themselves using the new authentication scheme. Timing data, error data and edit data was collected as in the other tasks.

Besides being assigned tasks, participants were administered an oral questionnaire wherein we asked them about their perceptions of the different tasks and their views on authentication tools like PINs and codebooks. Every participant was compensated with a gift worth 150 INR (3 USD) to participate in the study; the gifts were sponsored by our partner CKS India Pvt. Ltd.

4.3 Dependent Variables

For each participant and for each task performed by the participant, we computed two measures: the average time taken by the participant to complete that task (*task efficiency*) and the rate of error in performing it (*accuracy*). Task efficiency was measured by averaging the time taken by the participant on the last three trials of the task. Since the last three trials were error-free across all participants, this ensured a consistent measure of efficiency, free from learning effects and arbitrary influences that errors made while performing the tasks may have produced[5].

Measuring accuracy was a tricky affair. We distinguished between two types of errors made by participants: those that were made *before* the first successful trial, and those that were made *after* it. We ignored the former set of errors in our accuracy calculations, viewing them as having resulted from poor learning acquisition by the participant during the training protocol. The first successful trial by the participant was thus viewed as an indicator of his having acquired the ability to perform the task correctly without any external help. We defined the error rate as the ratio of the number of erroneous trials recorded from this trial onwards to the *total* number of trials performed (starting from the same trial). Put succinctly, for any task T and any participant P, the error rate with which P performed T was computed as:

Computing error rate of participant P performing task T:

1. Let n be the number of trials of T performed by P starting from the first successful trial up to task completion (that is, up to the point 3 error-free trials are complete). Clearly, $n \geq 3$ always.
2. Let e be the number of erroneous trials for T performed by P after the first successful trial has been performed.
3. Return error rate for P on task T as e/n.

[5] Indeed, during our experiments, we recorded several cases of unsuccessful signature entries wherein digits had been *omitted* from the signature. Incorporating the time taken in such trials would have unsuitably offset our efficiency measurements.

An alternate approach would have been to have each participant perform a fixed large number of trials, say N, starting from the first successful trial and compute the ratio of e – the number of erroneous trials in that period – to the number N. We chose the above approach to this alternative in order to avoid burdening the participants who made fewer errors (and who were expectedly larger in number) at the cost of those who made frequent errors (but were expectedly smaller in number). In our experiments, a majority of the participants did not make any errors after the first successful trial, but there were a few who performed more than 6 trials due to repeated errors.

4.4 Hypotheses

We posited the following hypotheses for our experiment:

(1) Participants, on average, require significantly more time to perform EKO signature formation than to perform new signature formation.

(2) Participants, on average, require significantly more time to perform new signature formation than to perform plain PIN entry.

(3) Participants' error rates, on average, are significantly higher in the case of EKO signature formation than in the case of new signature formation.

(4) More participants prefer the new authentication scheme to EKO's authentication scheme in terms of ease of use.

5. RESULTS

We were able to validate all the above hypotheses. In particular, we found that participants' task completion times were greater for EKO signature formation than for new signature formation and the latter, in turn, were greater than completion times for plain PIN entry. Both these gaps were statistically significant. We also observed a statistically significant gap between error rates for EKO signature formation and new signature formation, the latter being smaller. Finally, more participants reported to prefer the use of the new signature scheme over the use of EKO's scheme.

5.1 Efficiency

The mean task completion times for all tasks are given in table 1 and depicted in figure 5.

Table 1. Mean task completion time values (seconds)

Type of PIN	*Plain*	*EKO*	*New*
Simple	4.06	18.40	10.43
Complex	4.38	17.92	15.04

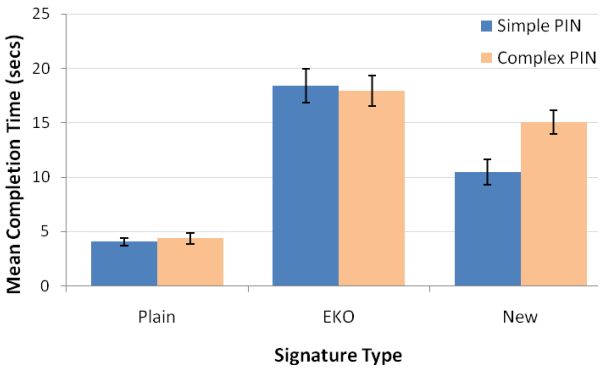

Figure 5. Mean task completion times for the 3 principal tasks performed by participants. Error bars denote standard error of the mean.

For measuring statistical significance, we performed a 2 (PIN complexity) x 3 (Signature type) within-subjects analysis of variance (ANOVA). Not surprisingly, there was a significant effect of PIN complexity ($F(1, 33) = 7.46$, $p < 0.01$); across signature types, the simple PIN was faster than the complex PIN. We also saw a significant effect for task type ($F(2, 33) = 105.3$, $p << 0.01$). As seen in Figure 5, plain PIN entry was faster than new signature formation, which, in turn, was faster than EKO signature formation. In addition, there was a significant interaction between the task type and the type of PIN ($F(4, 33) = 8.802$, $p << 0.01$). Figure 5 illustrates that for new signature formation, using a simple PIN made the task significantly faster than using a complex PIN. However, even when using the complex PIN for new signature formation, participants were faster than when using either of the PINs in EKO signature formation.

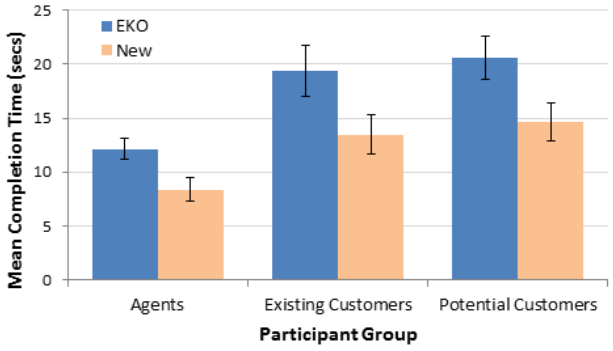

Figure 6. Comparison of mean task completion times for EKO signature formation and new signature formation across participant groups. Error bars denote standard error.

The difference between task completion times for EKO's scheme and the new scheme was significant across participant categories. We performed a paired sample t-test for the two sets of completion times for agents, existing customers and potential customers. In all three cases, we found significant differences. For agents, the difference was particularly stark ($t(2, 7) = 7.678$, $p << 0.01$) but even for the customers it was measurably significant ($t(2, 12) = 4.525$, $p << 0.01$ for existing, and $t(2,12) = 3.16$, $p << 0.01$ for potential customers). The means are depicted pictorially in figure 6.

It is plausible that the difference between task completion times we observed were due to the fact that the new scheme requires only 4 digits to be entered into the phone, as opposed to 10 digits for EKO's scheme. It is also plausible that the substitution-coding used in the new scheme is cognitively more complex than the juxtaposition technique of EKO: the ratio of task completion times for the two schemes is *not* the same as the ratio of the number of digits entered (which is just 4/10 = 0.4). Still, the difference in signature sizes seems significant enough to offset the greater cognitive load that the scheme might be placing on users.

The increase in efficiency of entering signatures in going from EKO's scheme to the new scheme seems to result in a near-equivalent increase in efficiency of creating transaction messages. In our experiments, participants, on average, took 43.13 seconds to create a transaction message when using EKO's scheme, as opposed to just 40.09 seconds when using the new one. (The transaction amount, recipient phone number and user PIN were held constant in both tasks.) Statistical significance could not be established, though, conceivably because the non-signature components of the transaction messages – which contained more data – had a greater influence on task completion times.

5.2 Accuracy

The mean error rates for all signature formation tasks are shown in table 2 and illustrated in figure 7. Note that for plain PIN entry, we recorded no errors whereas for the other tasks, an error rate of up to 6.5% was recorded.

Table 2. Mean error rate values.

Type of PIN	*Plain*	*EKO*	*New*
Simple	0	0.064776	0.006
Complex	0	0.054727	0.022549

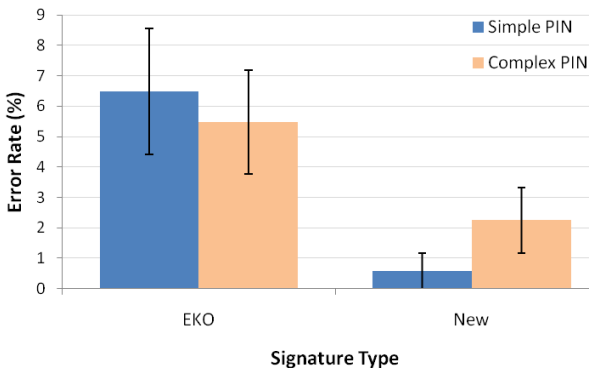

Figure 7. Mean error rates for the 3 principal tasks performed by participants. Error bars denote standard error of the mean.

To test the relative error rates, we performed a 2 (PIN complexity) x 2 (Signature type) within-subjects analysis of variance (ANOVA). Since performance on the plain PIN was essentially perfect, we only tested error rates for the two signature formations. We found a significant effect of task type ($F(1, 33) = 11.847$, $p << 0.01$), indicating that participants performed new signature formation task with significantly greater accuracy than the EKO signature formation task. There was no significant interaction effect of the type of PIN and the type of task, and

surprisingly, there was no significant main effect of the type of PIN on error rates either. The overall indication seems to be that error rates are significantly lower in the new scheme, and this holds independent of the type of PIN being utilized.

It could be that the lower error rates for the new authentication scheme are simply because there are fewer digits to enter into the phone in that scheme. We analyzed the types of errors participants were making in relation to the task type. Interestingly, 7 out of 34 errors that we observed for EKO signature formation were digit omissions and 5 were digit swaps (positions of two digits in the signature being interchanged), whereas in the case of new signature formation, *not a single digit omission or digit swap was encountered.* This observation points us to an interface design which supports minimal data entry from the user: in an already long and complex message that users are creating for conducting transactions, it is worth minimizing the amount of authentication information they enter into the phone without, of course, an unreasonable compromise on security. The new scheme is well in line with this design principle.

The difference in error rates for the two schemes seems to carry over to the task of creating full transaction messages as well: we recorded an error rate of 6.1% when participants used EKO's scheme in the transaction message, and 2.99% when they used the new scheme. The difference, however, was not statistically significant, plausibly due to effects of non-signature elements of the message. Also, in terms of the number of *edits* performed by participants during signature entry or transaction message entry, we did not observe any significant differences between EKO's scheme and the new one. Each participant, on average, performed 0.089 edits per signature task, and 0.191 edits per transaction task.

5.3 User Perceptions

Overall, participants seemed to prefer using the new scheme to the old one: 64.7% of the participants in our sample stated that they find the new scheme easier to use, while only 29.4% stated that they find it harder. The remaining 5.9% of the participants were neutral about the issue. Additionally, 7 out of 8 agents in our sample stated that the new scheme is easier to teach to the typical customer than the old one.

The preference statistics for the schemes across difference participant groups is shown in table 3.

Table 3. Participants' preferences for the two schemes.

Q. Which scheme do you prefer?	***Agents***	***Existing Customers***	***Potential Customers***
New	75%	69.23%	53.85%
EKO	25%	30.77%	30.77%
Neutral	-	-	15.38%

Participants expressed multiple reasons for their preference of the new scheme: *"… new signature scheme takes less time to understand as compared to old system (EKO)", "new one is easy to use because integration of PIN with 6 digits is difficult as compared to 'up-to-down' lookup task", "new signature scheme is easy as it involves typing only 4 digits", "only matching of numbers needs to be done, which is easy for me", "everything is given in the codebook, just needs to be looked up"*[6] etc. Two participants provided interesting comparisons of the cognitive load involved in the two schemes:

> In the new scheme, I need to lay less stress on my brain and more on my eyes, which is why it is easier to handle.

And the other one similarly said:

> I feel the new scheme is easier to use as it involves less physical work and more mental work whereas in the old scheme (EKO) both mental and physical work is required and in equal amounts.

It is worth remembering that almost a third of the participants did prefer EKO's PIN-entry method to what we propose and they provided some interesting justifications for this view, probably attributable to the increased cognitive demands of the substitution-coding. E.g., *".. encrypting the PIN takes time and makes it complicated, too", "I find the back-and-forth eye movement difficult", ".. old scheme is easier to comprehend", ".. in the old scheme, everything is one line, whereas here I need to look up and down; if the corresponding two matching digits (i.e. vertically-adjacent digits) are put in one box (instead of two), the new system will be easier to use."* etc. While some of these justifications could be attributed to individual perceptions, some others point to limitations in the current implementation of the new scheme and the visual depiction of the substitution code underlying it. As already stated, alternate designs of the codebooks are currently being considered and we hope to address some of the participants' concerns in future improved versions.

Interestingly, a few participants proactively expressed their views on the relative security offered by the two schemes. At least three participants in our sample stated that they find the new scheme more secure than the old one and articulated multiple reasons for this: *"the PIN is mixed up here and not written in plain, which means it is more secure", "it can be used in front of others even when they are watching";* the latter is a clear expression of participants' knowledge of – and paranoia towards – shoulder-surfing attacks. Shoulder-surfing attacks are a well-known possibility to developed-world bankers; we find it interesting that people even in the developing world are conscious about their possibility.

One participant drew a unique – and somewhat amusing – correlation between security and usability:

> The new signature scheme is more easy to use: It is more secure, therefore it is more easy.

If all bank customers in the world were like this participant, designing usable interfaces for secure banking would hardly be a challenge!

5.4 Discussion

Overall, the results from the study seem to indicate that the authentication scheme we have proposed in this paper fares better than EKO's scheme in terms of two key usability parameters – efficiency of usage and accuracy. The general preference of the

[6] All user quotes have been translated from Hindi, the language in which interviews were conducted. We have tried to minimize errors in translation to the extent possible.

target consumers also seems to be in favor of the new scheme. Most of the usability benefits of the new scheme can be attributed to one simple fact – the scheme requires users to enter fewer digits when authenticating themselves. There is a fairly general lesson here for user-centric design of *any* authentication protocol: reduce the amount of input you take from the user (to what is needed to avoid the most likely attacks), and your users will like your protocol!

Although the new scheme could not match plain PIN entry on either the efficiency or accuracy metrics, this disadvantage is probably compensated for by the greater PIN privacy the scheme offers. Indeed, if PINs are to be meaningfully deployed in any banking application, the bank must ensure secure methods to protect their privacy and given the challenges posed by current-day developing-world mobile networks, the possibility of building a secure system around plain PIN entry seems rather unlikely.

It is plausible that the new scheme will perform even better in practice than it did during the study. Since the scheme requires users to look up digits located at fixed positions in a number and since the number of lookups is fairly small (viz. four), the learning improvement with repeated usage may be greater than in EKO's scheme. As users become more accustomed to the lookup pattern corresponding to their PINs, the time required to execute the encryption and the chances of making mistakes are both likely to go down. However, verifying these claims rigorously will require a separate study and is outside the scope of the current work.

The fact that nearly 30% of our participant sample preferred EKO's scheme over ours is an issue worth understanding. While some of the difficulties that participants encountered in using our scheme could be attributed to limitations in the design of the codebooks, some others are perhaps due to the simple fact that the scheme is new and one that participants had no prior exposure to. Practice with the scheme will likely tilt user preferences further in its favor but, again, this claim requires independent investigation.

One issue that has not been suitably addressed by the current study is the usability barrier posed by storing and transporting codebooks. This is an issue not peculiar to mobile banking but fairly central to any token-based authentication system. There is a trade-off between security and usability in any system design and in a phone-based authentication solution – where the phones do not have sufficient capability to secure PINs – using physical tokens for PIN protection, and bearing the ensuing usability cost, seems nearly unavoidable[7]. One peculiar concern in EKO's context is the fact that the tokens here are made of paper which makes them more susceptible to damage and misuse. For example, in previous surveys of EKO's customers [29], it has been found that some users tend to write down their PINs in their codebooks to aid memory; such practice dilutes the purpose of using PINs altogether. (EKO is currently rigorizing its customer registration protocols to dissuade users from engaging in this practice.) There is also the issue of codebook theft and losses and EKO reported at least one codebook loss per month at the time of our study. This, again, is a challenge faced by every token-based authentication systems, although in the case of EKO the problem could be accentuated by the limited educational backgrounds of its customers.

Other limitations of our study include the lack of sufficient data to study gender effects on usability outcomes and the fact that we did not use Likert ratings to study user preferences quantitatively. In future field evaluations of the new scheme, we hope to address both these limitations suitably.

6. MORE RELATED WORK

There are several precedents to token-based user authentication in the literature and some of these are commercially deployed, too. The classic example is the RSA SecurID, a tool commonly used by companies for employee remote login. In SecurID-based authentication systems, like in ours, users must submit a fixed unique password as well as a one-time nonce (generated using special-purpose electronic dongles) to log in to a remote server. However, there are some key differences between our system and such schemes. For one, the passwords in our system are always numeric, which is the norm in banking transactions, and which, fortunately, also simplifies the task of encrypting them and enables it to happen in the user interface. For two, security of SecurID-based systems relies on a secure network which encrypts both the password and the nonce during transmission; in contrast, our solution is secure *even when run over insecure networks* because here, nonces themselves are used to encrypt the password prior to password-entry. Finally, SecurID uses system clocks for synchronizing nonces between server and user; such a facility is not part of our system but is being considered for future versions.

User-assisted encryption of PINs using one-time pads has some precedents in the banking world and in fact, there exists a commercial deployment of this concept. A company named Swivel [30] has developed a system for PIN-based 2-factor authentication for web banking applications which is very similar to ours. Like in our system, nonces are used to encrypt PINs via substitution-based coding before either of them is transmitted to the bank server. However, unlike our system, there is no storage mechanism for nonces; instead each nonce is communicated by the server to the user right before authentication. This communication must happen over a secondary channel of communication which, in the case of [30], is either mobile-based SMS or else an alternate web session. There are several challenges in implementing such a solution. For one, a secondary channel may not be available to every user and at every authentication instant. (This is particularly a problem if we apply the solution of [30] to mobile banking where the primary channel itself is the mobile network.) For two, even where the secondary channel is available, a suitable encryption interface may not be possible to implement. (Imagine, for example, transmitting entries of the codebook shown in figure 4, in an SMS.) For three, solutions like [30] are susceptible to phishing attacks – a phisher could potentially initiate an authentication session with the user, send an arbitrary nonce to him, acquire the encryption of the PIN under that nonce and use this information to recover the PIN. If nonces are stored locally by the user (like in our system), such an attack is harder to mount. To the best of our knowledge, no security analysis of Swivel's system is published in the literature, and no usability evaluation, even in developed-world contexts, is known.

Our work has implications for designing security systems in the presence of password-sharing practices. Given that password-

[7] The use of physical tokens could be avoided by doing biometric-based authentication, but such a system would have several associated costs. We are currently investigating other low-cost alternatives to token-based authentication solutions for mobile banking.

sharing (in particular, PIN-sharing) within select social circles is a prevalent practice even in the developed world [31], there is value in designing schemes which facilitate this behavior without significantly compromising security. A scheme like ours enables users to easily encrypt PINs before sharing them and thus can potentially reduce damage, if any, caused by PIN-sharing.

7. CONCLUSION

While the design of secure and usable authentication for banking applications is a well-studied problem in the developed world, applying the same solutions to developing-world mobile banking is a challenge, primarily due to the limited capacity of the phones available in these regions. Amongst all mobile banking providers in the world, EKO is unique in that it is striving to meet this challenge without resorting to network-level security protocols, without installing expensive biometric readers and while still enabling access to a low-literate user population. It is doing this by relying on a PIN-based solution and using simple, paper-based security tokens to encode PINs for privacy.

In this paper, we have demonstrated a security weakness in EKO's solution which causes the privacy of user PINs to be easily compromised. On the positive side, we have also shown an alternative solution which not only fixes this problem with EKO's scheme but also improves its usability and user-friendliness. This is an absolute win-win situation for user-centric security design – better security with better usability.

Our research has potential implications for banking in the developed world also. While ATM-based banking is claimed to offer secure 2-factor authentication, such claims have considerably weakened with the increasing incidence of skimming attacks in the recent past. A token-based solution like ours has clear advantages over what ATMs currently utilize, and it protects PINs irrespective of whether network encryption or anti-skimming measures are implemented by service providers. EKO's deployment of the concept in a developing-world context, accompanied by our usability evaluation of the improved system, provides evidence that such schemes are not only secure, but also usable and deployable at scale. We hope that our research will have a bearing on the design of future banking applications not only within the developing world, but also beyond it.

8. ACKNOWLEDGEMENTS

We thank Amit Gupta, Nishant Prakash, Sreedhar Reddy, Shalini Vishnoi and Shashank Shekhar from CKS India Pvt. Ltd. for conducting most of the field experiments for the usability study. Thanks to Aishwarya Ratan, Indrani Medhi, Prasad Naldurg and Raghav Bhaskar from Microsoft Research for participating in the early phases of this research. Finally, a special thanks to Anupam Varghese and his colleagues from EKO India Financial Services Ltd. for initiating the idea of the usability study and supporting it throughout its implementation.

9. REFERENCES

[1] M. Pickens, "Mobile money by the numbers," *http://technology.cgap.org/2009/06/04/mobile-money-by-the-numbers/*, Jun. 2009.

[2] G. Ivatury and I. Mas, "The Early Experience with Branchless Banking," *CGAP Focus Note 46*, Apr. 2008.

[3] Reserve Bank of India, "Policy Guidelines for issuance and operation of Prepaid Payment Instruments in India," *http://www.rbi.org.in/scripts/NotificationUser.aspx?Id=4953&Mode=0*, Apr. 2009.

[4] The Hindu News Online, "Fill in the b(l)anks," *http://www.thehindubusinessline.com/ew/2010/02/01/stories/2010020150050100.htm*, Feb. 2010.

[5] CRN News, "Biometrics becoming the norm for Aussie banking," *http://www.crn.com.au/News/161731,biometrics-becoming-the-norm-for-aussie-banking.aspx*, Nov. 2009.

[6] EKO India Financial Services Limited, *http://www.eko.co.in/*.

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

[8] WIZZIT, *http://www.wizzit.co.za/*.

[9] Globe - GCash, *www.globe.com.ph/gcash/*.

[10] O. Dunkelman, N. Keller, and A. Shamir, "A Practical-Time Attack on the A5/3 Cryptosystem Used in Third Generation GSM Telephony," *Cryptology ePrint Archive: Report 2010/013*, Jan. 2010.

[11] M. Paik, "Stragglers of the Herd Get Eaten: Security Concerns for GSM Mobile Banking Applications," *HotMobile 2010: The Eleventh International Workshop on Mobile Computing Systems and Applications*, Maryland: ACM, 2010.

[12] D. Hulton and Steve, "Cracking GSM," *Black Hat*, 2008.

[13] CNET News, "Bank of America plans to introduce wireless banking," *http://news.cnet.com/Bank-of-America-plans-to-introduce-wireless-banking/2100-1017_3-228389.html*, Jul. 1999.

[14] CelPay, *http://www.celpay.com*.

[15] Easy Paisa, *http://www.easypaisa.com.pk/easy-home.php*.

[16] High Beam Research, "More Than 10% Of Kenya's GDP Now pass through the M-Pesa Mobile Banking Service," *http://www.highbeam.com/doc/1G1-193984464.html*, Feb. 2009.

[17] B. Lorica, "Mobiles and Money in the Developing World," *Release 2.0*, Apr. 2009.

[18] RSA SecureID, *http://www.rsa.com/node.aspx?id=1156*.

[19] A. Sharma, L. Subramanian, and D. Shasha, "Secure Branchless Banking," *ACM SOSP Workshop on Networked Systems for Developing Regions (NSDR)*, Montana: ACM, 2009.

[20] Grameen Koota, *http://www.grameenkoota.org*.

[21] A. Das, "Audio Visual Person Authentication by Multiple Nearest Neighbor Classifiers," *Advances in Biometrics*, Springer Berlin/Heidelberg, 2007, pp. 1114-1123.

[22] FINO Ltd., *http://www.fino.co.in/*.

[23] A Little World, *http://www.alittleworld.com*.

[24] S. Panjwani, P. Naldurg, and R. Bhaskar, "Analysis of Two Token-Based Authentication Schemes for Mobile Banking," *Microsoft Research Technical Report*, 2010.

[25] S C Magazine, "U.S. businesses face skimming fraud increase," *http://www.scmagazineus.com/us-businesses-face-skimming-fraud-increase/article/168793/*, Apr. 2010.

[26] V. Roth, K. Richter, and R. Freidinger, "A PIN-Entry Method Resilient Against Shoulder Surfing," *Proc. of the 11th ACM conference on Computer and communications security (CCS)*, Washington DC, USA: ACM, 2004, pp. 236-245.

[27] M. Kumar, T. Garfinkel, D. Boneh, and T. Winograd, "Reducing Shoulder-surfing by Using Gaze-based Password Entry," *Proc. of Symposium On Usable Privacy and Security (SOUPS)*, Pittsburgh, USA: 2007.

[28] A. De Luca, M. Denzel, and H. Hussman, "Look into my Eyes! Can you guess my Password?," *Proc. of Symposium on Usable Privacy and Security (SOUPS)*, Mountain View, USA: 2009.

[29] A. Ratan and I. Medhi, "EKO: An Evaluation Study," *Microsoft Research Technical Report*, 2008.

[30] Swivel Authentication Systems, "PINsafe Multifactor Authentication Solution," *http://www.swivelsecure.com/*, 2003.

[31] S. Singh, A. Cabraal, C. Demosthenous, G. Astbrink, and M. Furlong, "Password Sharing: Implications for Security Design Based on Social Practice," *Proc. of CHI*, San Jose: ACM, 2007.