CROO: A Universal Infrastructure and Protocol to Detect Identity Fraud (Extended Version)*

D. Nali P.C. van Oorschot

July 14, 2008

Abstract

Identity fraud (IDF) may be defined as unauthorized exploitation of credential information through the use of false identity. We propose CROO, a universal (i.e. generic) infrastructure and protocol to either prevent IDF (by detecting attempts thereof), or limit its consequences (by identifying cases of previously undetected IDF). CROO is a capture resilient one-time password scheme, whereby each user must carry a personal trusted device used to generate one-time passwords (OTPs) verified by online trusted parties. Multiple trusted parties may be used for increased scalability. OTPs are generated and verified for any desired user transaction, and can be used regardless of the transaction's purpose (e.g. user authentication or financial payment), associated credentials, and online or on-site nature; this makes CROO a universal scheme. OTPs are not sent in cleartext; they are used as keys to compute MACs of hashed transaction information, in a manner allowing OTP-verifying parties to confirm that given user credentials (i.e. OTP-keyed MACs) correspond to claimed hashed transaction details. Hashing transaction details increases user privacy. Each OTP is generated from a PIN-encrypted non-verifiable key; this makes users' personal devices resilient to off-line PIN-guessing attacks. CROO's credentials can be formatted (hence used) as existing user credentials (e.g. credit cards or driver's licenses).

1 Introduction

We informally define identity fraud (IDF)[1] as unauthorized exploitation of extracted credential information (e.g. identification passwords, driver's licence numbers, and credit card numbers) involving some form of impersonation or misrepresentation of identity. A 2005 survey [51] reported that over 9 million Americans (i.e. one in 23 American adults) were IDF victims in 2004, corresponding to a total annual cost of $51.4 billion and a median cost of $750 per victim. The motivation behind IDF is multifaceted and the possible damages are diverse. These damages include loss of privacy, worry and fear, financial loss, time loss, denial of service,[2] and public discredit [5].

In the academic literature, there are relatively few proposals addressing IDF. Most focus on prevention of credential information extraction. (See, e.g. [9, 13, 29, 39, 52], for countermeasures to phishing or key logging.) We are aware of only one non-application-specific academic proposal addressing generic IDF [61], which, as presented, has limitations including restriction to on-site (vs. online) transactions and loss of user location privacy (users are geographically tracked).

In this paper, we focus on IDF (as defined above). More precisely, while such impersonation attacks may be committed under fictitious identities (e.g. identities of non-existing people to conceal involvement in illegal activity), we focus on IDF involving real people's identities, in large part because we wish to focus on IDF that consumes real people's time. We also include consideration of IDF involving newly created credentials (e.g. credit, health, and building access cards) obtained by fraudsters in their victims' names, because this type of IDF currently seems difficult to detect. Our focus is on the *generic* IDF problem, and we seek a universal IDF solution, i.e. one which works for both remote and on-site transactions, and is neither application-specific, nor restricted to instances (of the generic IDF problem) associated with one class of credential tokens (e.g. credit cards). Credential-specific solutions are potentially what individual applications' (e.g. credit card) vendors are

*Contact Author: deholo@gmail.com. This technical report updates an earlier report [47] in several aspects, including a modification to the CROO protocol. The earlier report contains an AVISPA analysis of an earlier protocol, and an IDF model. A shortened version of this report will appear in the proceedings of ESORICS 2008.

[1] We prefer this term over "identity theft" (IDT), although both have often been used [18, 22, 34]. The term theft seems to suggest that victims are "deprived" of their identity, which is not always true, nor our focus.

[2] IDF victims have been arrested due to fraud committed by their impersonators under the victims' names [12].

likely to propose; we believe end-users will find universal solutions both more usable (when considered across all applications), and less costly in terms of personal time. One might also argue that, for overall economic reasons, an IDF solution detecting driver's license and health/debit/credit card-based forms of IDF is more likely to be adopted and accepted by card bearers and state and financial institutions than solutions which only detect one of these forms of IDF.[3]

While we deal with architectural problems associated with the design of privacy-preserving credential management systems (cf. [7]), our primary focus is not the privacy aspect of such systems, but their fraud detection capability. Similarly, we do not aim to solve the bootstrap problem of human identification at the time when credentials are issued to people. Instead, we assume that trusted parties exist that can identify legitimate users (e.g. using out-of-band mechanisms), and we focus on the detection of fraudulent uses of credentials.

We propose a universal infrastructure and protocol for IDF detection, which we call CROO (Capture Resilient Online One-time password scheme). Each user must carry a personal device used to generate one-time passwords (OTPs) verified by online trusted parties. These OTP generation and verification procedures are universal, in the sense that they can be associated with any user transaction, regardless of the transaction's purpose (e.g. user identification, user authentication, or financial payment), associated credentials (e.g. driver's license or credit card), and online or on-site (e.g. point-of-sale) nature. For increased scalablity, multiple OTP verification parties may be used (see §2.5). OTPs are not sent in cleartext; they are used as keys to compute MACs of hashed unique transaction information (e.g. list of bought items). This allows OTP-verifying parties to confirm that given user credentials (i.e. OTP-based MACs) correspond to claimed hashed transaction details. Hashing transaction information increases user privacy. Online OTP-verifying parties detect IDF when OTPs of received user credentials or the associated transaction information do not have expected values. Each OTP is generated from a high-entropy non-verifiable text [37] encrypted using a key derived from a user-chosen PIN; hence, possession of a user's personal device (or clone thereof) does not suffice to confirm guesses of the associated PIN, to recover the associated non-verifiable key, and generate correct OTPs. Since OTPs can only be verified by online parties, the proposed scheme turns off-line PIN guessing attacks against stolen or cloned personal devices into online OTP-guessing attacks that can be easily detected by online parties.

An interesting aspect of CROO is that it provides means to both prevent IDF (by detecting IDF attempts), and limit its consequences when sophisticated IDF attacks have bypassed the aforementioned preventive measures. The latter goal is achieved by identifying cases in which undetected IDF victims use any of their credentials. Limiting the consequences of IDF is of use when a fraudster has acquired a user's PIN, stolen the user's personal device, and used the device to generate correct OTPs for unauthorized transactions. Another interesting aspect of CROO is that it requires minimal changes to existing credential processing protocols: users continue to interact with relying parties; relying parties continue to interact with users and card issuers; and the proposed OTP-based user credentials can be customized to follow existing formats (e.g. the credit card numbering format). To achieve this, CROO requires card issuers to interact with both relying parties and proposed online OTP-verifying parties. For space and processing speed efficiency, CROO employs MACs (vs. encryption or public-key signature) to generate user credentials. From a practical standpoint, users employ personal devices to generate OTP-based credentials which can be used in the same way existing credentials are used. To generate OTPs, personal devices must receive transaction details (e.g. a dollar amount and relying party's identifier). These details can be communicated either by manual keyboard entry, or via short-range wireless communication with local terminals (e.g. by waiving the personal device before a transceiver linked to a local terminal).

CROO relies on malware-free personal devices in which secrets used to generate OTPs are stored. As others [21, 44], we believe that, in the near future, a subset of deployed personal devices will meet this requirement, possibly as a result of initiatives such as the Trusted Computing Group [2].

Contributions. The proposal of a universal infrastructure and protocol for addressing IDF is our main contribution. *Universal* here means designed to be simultaneously used with multiple classes of user transactions, i.e. regardless of transactions' applications, on-site or remote nature, purposes, attributes, and associated credentials. We analyze the proposed scheme using criteria grouped into categories of usability, privacy, fraud detection, and communication security. The diversity and number of these criteria reflect the challenge in designing universal IDF detection systems.

Outline. §2 describes CROO. §3 presents evaluation criteria, and analyzes CROO. §4 discusses related work. §5 concludes the paper.

[3]Both health cards (HCs) and drivers' licenses (DLs) can be used to commit IDF. Stolen or cloned HCs can be misused to gain access to medical services, and DLs exploited to obtain illegitimate new credit cards, loans, and merchandize.

2 Proposed Infrastructure and Protocol for IDF Detection

This section proposes `CROO`, a universal infrastructure and protocol for IDF prevention and/or after-the-fact detection for consequence limitation. It is intended to be simultaneously usable with multiple classes of applications and credentials, for both online and on-site transactions.

2.1 Fundamental Definitions

Before presenting `CROO`, we define a few terms related to IDF. In this paper: *identity* (ID) denotes a collection of characteristics by which a person is known (an individual may have more than one identity); *identifier* refers to any label assigned to an identity to distinguish this identity from other identities; *credential information* (*cred-info*) denotes information (or a piece thereof) presented by a party to either gain privileges or goods, or to support the veracity of an identity-related claim made by this party; and *credential token* (*cred-token*) refers to an object (tangible or electronic) on which cred-info is recorded.

2.2 Architectural Components

Parties. Let I be a party that issues cred-tokens and authorizes, when needed, the execution of operations associated with cred-tokens issued by I. (For example, I may be a credit card company that issues credit cards and authorizes payments made with these cards.) Let F be a party that monitors the use of cred-tokens, and can assign identifiers to a person U. (In some practical instantiations, F may be a sub-component of party I, and/or the two may be co-located.) Assume that I issues a cred-token C_U to U, and let R be a party that provides goods or services to any person or organisation A, when the following conditions are satisfied: (1) A presents to R either certain cred-tokens (e.g. a credit card) or pieces of cred-info (e.g. a credit card number and a name); and (2) either the items presented to R grant A required privileges, or confirm that A has required attributes (e.g. is of a certain age).

Personal Device. U acquires a personal trusted computing device D_U equipped both with an input/output user interface and capability to communicate via a standard short range wireless (SRW) channel (e.g. an NFC-[4] or Bluetooth-enabled cell phone, if suitable as a trusted computing platform, or a small special-purpose device usable for multi-application IDF prevention and detection.) Any communication between D_U and F, R, or I is over the SRW channel. When R is an online party in a web-transaction (rather than a physically present point of sale), then communication between D_U and R combines SRW communication between D_U and a PC, and Internet-based communication between this PC and R. If D_U uses NFC to communicate with other devices, then U simply needs to waive D_U before these devices for communication to take place. As a fall-back measure, when no electronic (e.g. NFC-based) SRW channel can be used by D_U to communicate with other devices, a manual or oral communication channel may be used, whereby U manually or orally (e.g. in the case of phone-call-based transactions) communicates information needed or output by D_U.

2.3 IDF Detection Protocol

The IDF detection protocol consists of an Initialization protocol and a Transaction protocol. The notation of Table 1 is henceforth used. Table 2 summarizes the Transaction protocol.

Initialization.

1. I provides U with C_U.
2. U appears before (or engages in an audiovisual phone conversation with) F to allow F to verify that U is who she claims to be.[5] This is done using standard (e.g. out-of-band) techniques. If F is not convinced of U's identity, the `Initialization` procedure is aborted. Otherwise:
3. F generates and provides U with (s_U,$k^{(n)}$, n, ID_U). F also sets an ID_U-specific counter i to 0.
4. U chooses and memorizes a PIN p_U, and inputs (s_U,$k^{(n)}$, n, ID_U) in D_U. D_U generates a d_2-bit nonce q, and computes $\{s_U, k^{(n)}, q\}^{\hat{p_U}}$ (i.e. symmetrically encrypts (s_U, $k^{(n)}$, q) with a key $\hat{p_U}$ derived from p_U and a secure symmetric encryption scheme e.g. AES-128 in CBC mode). Then D_U stores the ciphertext locally, sets to 0 a counter i, and erases p_U and $\hat{p_U}$ from its memory.
5. U sends ID_U to I, and indicates to I that F monitors C_U, and C_U must be paired with ID_U. U also provides I with D_U's number if D_U is a mobile phone. I links ID_U, C_U, and D_U's number if applicable.

[4] NFC [19] enables wireless communication between devices in very close (e.g. less than 10cm) proximity.

[5] Instead, U may visit a trusted representative of F. However, for simplicity, we henceforth assume that U visits F.

Symbol	Explanation
$\{d_i\}_{i=1}^{6}$	Length parameters. E.g. $d_1 \geq 4$, $d_2 = 160$, $d_3 = 128$, $10 \geq d_4 \geq 5$, $d_5 = 36$, $d_6 = 72$.
n	Number (e.g. 10,000) of cred-tokens or pieces of cred-info monitored by F per user. When F has monitored n transactions for U, Steps A and B of the Fraud Recovery protocol are executed.
C_U	Cred-token issued to U by I.
ID_U	Unique temporary identifier assigned to U by F, e.g. a bit string, or U's name and postal address.
p_U	d_1-digit PIN chosen and memorized by U.
s_U	d_2-bit secret random salts generated by F.
$\hat{p_U}$	Symmetric key derived from p_U (e.g. first d_3 bits of $h(p_U)$).
h	Cryptographic hash function (e.g. SHA-1) with d_2-bit image elements.
f	MAC (e.g. SHA-1-HMAC) with co-domain elements of same bit length as s_U.
$k^{(j)}$	j^{th} d_2-bit one-time password. $k^{(n)}$ is a random secret d_2-bit string generated by F. $k^{(j)} = h(s_U, k^{(j+1)})$ for $j = n-1, n-2, \cdots, 0$.
z	Transaction details (e.g. timestamp, dollar value, and R's 10-digit phone number).
G	Function which, given a d_2-bit string (equal to $f_{k^{(i)}}(h(z))$ in the Transaction Protocol), constructs a well-formatted d_6-bit string allowing R to determine the issuer I to which $G(x)$ is intended, and such that $\lvert G(\{0,1\}^{d_2})\rvert$ is d_5 bits. E.g., if (a, b) denotes the concatenation of two strings a and b, one can define $G(x) = (y_1, y_2, y_3, y_4)$, where y_1 is a 6-digit identifier of I, y_3 is a single-digit check code, and y_4 is a 3-digit verification code such that $(y_2, y_4) = x \ mod \ 10^{11}$ and (y_1, y_2, y_3) is a syntactically-valid credit-card number (CCN); in this case, $G(x)$ is akin to the concatenation of a 15-digit CCN (y_1, y_2, y_3) with a 3-digit verification code y_4. In the transaction protocol, $G(x)$ is either manually input by U in a local terminal, or automatically transferred thereto via NFC as U waives D_U before a receiver.
S_z	Fraud status issued by F for the transaction associated with z.
A_z	Receipt issued by I concerning the transaction associated with z.

Table 1: Notational Overview

6. I and F (respectively R and I) acquire cryptographic material required to establish secure channels between each other (e.g. by exchanging each other's public-key certificate). Throughout the paper, *secure channels* denote communication channels providing confidentiality, integrity, bi-directional authenticity, and message-replay protection for a chosen time frame (e.g. by storing cryptogrpahic hashes of all messages received in the last hour).

Transaction.

1. R sends z to D_U.
2. D_U displays z to U, and U inputs p_U in D_U. Let i be the value stored by D_U. D_U computes $k^{(i)}$ (see Table 1) and $v = G(f_{k^{(i)}}(h(z)))$. Then, D_U increments i, and sends (ID_U, v) to R.
3. Upon receiving (ID_U, v), R sends (ID_U, v, z) to I, over a secure channel.
4. Upon receiving (ID_U, v, z), I sends $(ID_U, h(z), v)$ to F, over a secure channel.

U	D_U	R	I	F	Messages Sent
1.	| ← |				z
2. | ← |					z
2. | → |					p_U
3.	| → |				(ID_U, v), where $v = G(f_{k^{(i)}}(h(z)))$
4.		| → |			(ID_U, v, z)
5.			| → |		$(ID_U, h(z), v)$
6.			| ← |		$(ID_U, h(z), S_z)$
7.		| ← |			$(h(z), A_z)$

Table 2: Transaction Fraud Verification Protocol

5. Upon receiving $(ID_U, h(z), v)$, F uses ID_U to retrieve information required to compute $k^{(i)}$,[6] and checks wether $v = G(f_{k^{(i)}}(h(z)))$. Then F computes the fraud status variable S_z as follows: (a) if $v = G(f_{k^{(i)}}(h(z)))$, F sets $S_z = 0$; (b) if $v = G(f_{k^{(i-j)}}(h(z)))$ for some integer j such that $1 \leq j \leq d_4$, then F concludes that U has been impersonated, and sets $S_z = 1$. (c) otherwise, F proceeds as follows: (c1) if the values v of tuples $(ID_U, h(z), v)$ received by F have been incorrect for more than a small number of times (e.g. 5 or 10), within a F-chosen time period, then F concludes that U's cred-tokens are currently under attack, and F sets $S_z = 2$;[7] (c2) otherwise, F sets $S_z = 3$. Then F sends back $(ID_U, h(z), S_z)$ to I over the channel from which $(ID_U, h(z), v)$ was just received.
6. Upon receiving $(ID_U, h(z), S_z)$, I uses ID_U to retrieve C_U, and proceeds as follows: if $S_z = 0$, I uses C_U to process the transaction request (ID_U, v, z) according to I-chosen business rules (e.g. z includes a very recent time stamp and sufficiently low dollar amount, or, when `CROO` is used for authentication only, z is an authentication request including a nonce), and sets $A_z = 0$; if $S_z = 1$, I sets $A_z = 1$, and follows a predefined procedure (e.g. I may directly notify U by calling D_U if D_U is a mobile phone); if $S_z = 2$, I sets $A_z = 1$, and follows another predefined procedure (e.g. I may temporarily declare all uses of cred-info associated with ID_U as fraudulent); if $S_z = 3$, I sets $A_z = 1$, and follows yet another predefined procedure (e.g. I may not do anything). Then I sends $(h(z), A_z)$ to R using the channel from which $(ID_U, h(z), v)$ was sent.
7. Upon receiving $(h(z), A_z)$, R proceeds as follows: if $A_z = 0$, R provides U with the expected goods or services; otherwise, R notifies U that the transaction was not successful, and issues a receipt to U mentioning that the given transaction failed.

Fraud Recovery. Upon suspecting that she has been impersonated,[8] U either phones or goes to F in person. Then, the following steps A and B are executed. (A) F verifies U's claimed identity (e.g. using out-of-band procedures),[9] and proceeds as follows. (B) F resets U's counter i to 0; U obtains new (s_U,$k^{(n)}$, n) from F, and chooses and memorizes a new p_U; D_U generates a d_2-bit nonce q, computes $\{s_U, k^{(n)}, q\}^{\hat{p_U}}$ and stores the result on D_U; D_U also sets to 0 the counter i, and erases p_U and $\hat{p_U}$ from its memory.

2.4 Concrete Examples of `CROO`

Credit Card. A real-world instantiation of `CROO` could be as follows: I is a credit card company; R is an online merchant; U is a legitimate customer of I to whom I issues a credit card C_U; F is a credit bureau; and D_U is a cell phone equipped with a software application facilitating web-based online commerce via PCs. The transaction phase of `CROO` would then look like this: U accesses R's web site from a PC, and selects a list of items to buy; R sends the associated transaction details to U's PC; these details (e.g. dollar amount and R's identifier) are manually input by U into D_U; U inputs a PIN into D_U; D_U displays (ID_U, v), which U manually inputs into her PC and this 2-tuple is sent to R via SSL; R contacts I via SSL and I contacts F in the same way; if C_U's credit limit exceeds the charges of the current transaction, and if, according to F (as explained in the transaction protocol), the current transaction is non-fraudulent, then I sends R a promise of payment for an appropriate amount of money; and R notifies U, via the web and U's PC, that the transaction has been approved.

Driver's License. As a second example, `CROO` can be instantiated with the following parties: I is a state agency that issues drivers' licences; R is a bank; U is a person to whom I issues a driver's licence C_U; F is a state agency that specializes in the detection of fraud involving state-issued cred-tokens. (When validation of driver's licence information (e.g. for credit card issuing) does not currently involve online check with a trusted party, this second instantiation of our (online) proposal may be used to better detect driver's license-related IDF.)

Student ID. A third instantiation of `CROO` has the following parties: I is a university that issues facility access cards to its students; U is an employee of I; R, F, and I are modules within the same entity; C_U is a facility access card issued to U by I; D_U is a cell phone communicating with electronic door locks via NFC.

[6] e.g. i and $k^{(n)}$, or i and $k^{(i+d_4)}$ if $i + d_4 \leq n$ and $k^{(i+d_4)}$ was stored by F to speed up the computation of $k^{(i)}$.

[7] Step 5(c1) requires F to store a counter indicating the number of times the associated condition has been satisfied over a chosen time period. This counter must be set to 0 when S_z is set to 0 or 1 while processing a request associated with ID_U.

[8] Such suspicion may come to U from reviewing personal transaction reports.

[9] If fraud recovery is initiated more than a predefined number of times in a given time-frame, F may engage in more thorough authentication of U (e.g. via in-person thorough interviews by representatives of F).

2.5 Extensions

CROO is flexible with respect to the number of credential issuers I and the number of fraud detection parties F. In other words, U may have cred-tokens issued by different parties I, and these parties may rely on different fraud detecting parties F. For example, fraud detecting parties may be peculiar to particular applications or contexts (e.g. financial or government-oriented services). In some cases, however, it may be simpler to associate all the cred-tokens of a user with a single fraud detecting party, even though this party might not be the same for all users (e.g. for scalability purposes). The advantage of using a single fraud detecting party for all cred-tokens of a user is that when fraud is committed with any of this user's cred-tokens, this instance of fraud is detected the next time the user utilizes any of its cred-tokens. This is due to the fact that each one-time password is not bound with a particular cred-token, but with a user and the party that validates this OTP. In other words, one-time passwords are used across cred-tokens and cred-info thereon. Another extension of CROO consists in asking users (say U) to memorize different PINs for different groups of cred-tokens; if a PIN is guessed by an attacker, the cred-tokens associated with PINs that have not been guessed may still be used by U, and the OTPs associated with the non-guessed PINs are not temporarily declared as fraudulent.

3 Analysis of Proposed Solution

This section discusses evaluation properties for analysis and comparison of the proposed CROO protocol (henceforth denoted S, for scheme) with others. We are primarily interested in conveying an understanding of S's usability, privacy, and security characteristics (using practical criteria presented in §3.1), rather than algebraically "proving" the security of S. In §3.2, we discuss security- and privacy-related requirements of CROO. Devising realistic mathematical models and formal proofs which provide tangible guarantees in real-world deployments remains a challenge for us and others [31, 30]. However, we present in §3.3 a preliminary mathematical security analysis of a simplified version of CROO.

3.1 Comparative Evaluation Criteria for Universal ID Fraud Solutions

We aim to provide criteria that can be used to evaluate the effectiveness of the proposed IDF detection scheme. We consider criteria under four categories: usability, privacy (i.e. ability of users to control access to their cp-info), fraud detection capability (i.e. capability to detect IDF attempts or cases in which IDF has been committed without being detected), and communication security (e.g. protection against man-in-the-middle attacks). Presented below, these criteria are not exhaustive, but rather what we hope is a useful first step towards an accepted set of criteria to evaluate universal IDF solutions.

The following notation is used: I is any legitimate credential issuer; U is a user (person); x_U is a cred-token or cred-info issued by I to a person believed to be U; $x_U{}^*$ denotes x_U and/or any clones thereof; and R is a (relying) party whose goal is either to verify claims made by, or provide goods/services to, any party A, provided A demonstrates knowledge of appropriate secret information, or shows possession of certain cred-tokens or cred-info that are both valid and not flagged as fraudulent. Moreover, terms denoted by † can further be qualified by "instantly" or "within some useful time period".

Notation ✓ (resp. ✗) indicates that S meets (resp. does not meet) the associated criterion. Notation ✓✗ indicates that the associated criterion is partially met by S. Details of the evaluation claims are presented inline, for each evaluation criterion.

Usability Evaluation Criteria

- ✓ **U1.** *No Requirement to Memorize Multiple Passwords.* S does not require U to memorize cred-token-specific or application-specific passwords. Evaluation: U is required to memorize only one PIN, instead of many issuer or application-specific passwords.
- ✓✗ **U2.** *No Requirement to Acquire Extra Devices.* S does not require U to acquire extra devices (e.g. computers, cell phones, memory drives).[10] Evaluation: If U carries a NFC-enabled cell phone or PDA, then U does not need to acquire an extra portable device; otherwise, U needs to acquire and carry D_U.
- ✓✗ **U3.** *No Requirement for Users to Carry Extra Devices.* S does not require U to carry extra personal devices (e.g. cell phone). Evaluation: See U2.
- ✓✗ **U4.** *Easy Transition from Current Processes.* S does not require U to significantly change current processes to which U is accustomed (e.g. by not requiring extra mental or dexterous effort from U). For example, U is likely used to entering a PIN when using bank cards (vs. having an eye scanned). Evaluation: S requires U to enter a PIN when she uses C_U; this is assumed not to cause too drastic a

[10] S may require U to load new software on an existing general-purpose device (e.g. cell phone).

change from current processes (cf. debit card use); moreover, S recommends the use of a standard wireless communication channel for interaction between D_U and either R or a local PC communicating with R. However, some instantiations of S might implement this interaction through manual copy of v (i.e. a string akin to the concatenation of a CCN with a 3-digit verification code) by U.

✓ **U5.** *Support for Online Transactions.* S detects instances of attempted and/or committed but previously undetected IDF for online (e.g. web) transactions. Evaluation: If R or this local PC can communicate over the standard channel, then S can be used both for on-site and on-line transactions; (as presented, [61]'s proposal is restricted to on-site transactions;) S is therefore suitable for both mobile and fixed users.

✓ **U6.** *Support for On-Site Transactions.* S detects instances of attempted and/or committed but previously undetected IDF for on-site (e.g. point-of-sale) transactions. Evaluation: See `U5`.

✓ **U7.** *Convenience of Fraud Flagging Procedures.* When IDF has been detected (e.g. by a user or system), S provides a convenient mechanism to flag the appropriate cred-tokens as fraudulent. For example, S may enable U to interact with only one party to flag, as fraudulent, any of her cred-tokens. Evaluation: See `U5`.

✓ **U8.** *Suitability for Fixed Users.* S can be used by fixed users (i.e. who carry out transactions from a constant geographic location). Evaluation: See `U5`.

✓ **U9.** *Convenience of Fraud Recovery Procedures.* When U suffers IDF, S allows U to easily recover. For example, S may enable U to interact with only one party to obtain new cred-tokens that can be used thereafter, without having to obtain new cred-tokens from a number of credential issuers. Alternatively, S may enable U to interact with only one party that allows her to both continue to use her cred-tokens, and have the assurance that the use of any clones of her cred-tokens will be detected as fraudulent. Evaluation: U needs to interact with only one party (i.e. F) in order to recover from IDF, and U does not then need to change her cred-tokens (she only needs to send to F new initial OTP information).

✗ **U10.** *Support for Transactions Involving Off-line Relying Parties.* S detects instances of attempted and/or committed but previously undetected IDF even if R is not able to communicate, in real time, with other parties (e.g. I and F). Evaluation: S does not work if R cannot interact with I and F at each transaction.

✗ **U11.** *Support for Use of Multiple Credentials in a Single Transaction.* S enables the use of multiple pieces of cred-info in a single transaction. Evaluation: In its current form, S does not support this feature.

Privacy Evaluation Criteria

✓✗ **P1.** *No Disclosure of User Location.* S does not disclose U's location information, e.g. to multiple entities, or an entity that shares it with other parties. Evaluation: S does not require the use of user-location information in order to detect IDF (unlike [61]'s proposal); this does not mean, however, that U's location information cannot be known (e.g. by D_U's call carrier if D_U is a GPS-enabled cell phone); nonetheless, the fact that U's location information is not required to detect IDF might decrease the odds that this information be collected; (hence, S is potentially less privacy invasive than the scheme in [61]).

✓✗ **P2.** *No Disclosure of User Activity.* S does not disclose transaction details regarding U's activity (e.g. what U has bought, and when or where this was done). Evaluation: I knows C_U; however, S is designed in such a way that z is not revealed to F, and C_U is revealed neither to R nor F.

✓ **P3.** *No Disclosure of User Capabilities.* S does not reveal what hardware or software capabilities (e.g. digital camera or printer) U has. Evaluation: S does not require R to reveal to I private information such as lists of items bought by U and private information recorded on monitored cred-tokens; instead, R may include in transaction details only information that is necessary, e.g. a dollar value and an identifier of R.

✓ **P4.** *No Disclosure of User's Private Information.* S does not reveal private (e.g. medical or financial) user information. Evaluation: See `P3`.

Fraud Detection Evaluation Criteria

✓ **D1.** *Determination of Credential Use.* U and I know† when x_U* is used. Evaluation: For T to be successfully completed, I must send its approval thereof to R; consequently, I detects each use of C_U, and, if necessary, I notifies U accordingly (e.g. via D_U or postal mail).

✓ **D2.** *Control on Credential Use.* U and I can control† the use of x_U* (i.e. approve or reject each use thereof). Evaluation: For the same reason in `D1`, I controls the use of C_U (i.e. can reject or approve of each use of C_U); if necessary, I may consult U (e.g. via D_U) before approving of T; h Hence, S may be instantiated in such a way that U controls each use of C_U.

✓ **D3.** *Detection of Illegitimate Credential Holder.* When x_U* is presented to R, then U, I, and R can determine whether x_U*'s holder is authorized to hold x_U*. This credential holder legitimacy check might be based on the possession of a specified token, the knowledge of a memorized secret, the presentation of inherent biometric features, the proof of current geographic location, or some other criterion. Evaluation:

In order for F to accept $k^{(i)}$ as non-fraudulent (i.e. in order for an adversary A to generate the correct $k^{(i)}$), A must have an authentic copy of $\{s_U, k^{(n)}, q\}^{p_U}$ and a correct guess of both p_U and i; since these three pieces of information are not easily obtained without successive guesses of p_U and the cloning or capture of the mobile device D_U, A is not likely to generate the correct $k^{(i)}$ within an undetected number of guesses (e.g. fewer than 5 trials); if A is able to clone D_U, then i and $\{s_U, k^{(n)}, q\}^{p_U}$ can be obtained, but any significant number of wrong guesses of p_U is detected by F; F notifies I of any fraudulent use of C_U (or m_{C_U}), and, if necessary, F also notifies U (e.g. via D_U); hence, F, I, R, and potentially U can detect illegitimate credential holders.

✓✗ **D4.** *Determination of Credential Use Context.* U and I can determine† in which context (e.g. R's identity, network location, and geographic location) x_U* is used.[11] Evaluation: I is not necessarily able to reliably determine the context of R (e.g. R's geographic or network location); however, if I and R interact via a broker that is trusted by I, then this broker may be able to provide I with reliable contextual information about R; (note that certain cred-tokens may not be used with relying parties associated with certain contextual attributes, e.g. foreign relying parties;) if necessary, I can communicate this contextual information to U (e.g. via D_U).

✓✗ **D5.** *Verification of R's Entitlement to View Credential.* U and I can determine† whether R is a party to which x_U* is authorized to be shown for specified purposes (e.g. the delivery of cred-tokens, goods, or services). Evaluation: See `D4`.

✓ **D6.** *Entitlement Verification of Credential Holder's Claimed ID.* R and I can determine† whether x_U* is associated with its holder's claimed identity.[12] Evaluation: Neither I nor F is able to determine, with full assurance, the true identity of the issuer of v. R may verify U's claimed identity by requesting C_U from U, requiring that a photo of A appear on C_U, and making sure $A_z = 0$; however, if A is an attacker holding C_U, A may tamper C_U by substituting U's photo with A's; nevertheless, A is not likely to be able to use C_U successfully, because A is not likely to issue a correct $k^{(i)}$ before being detected (and blocked) by F.

✓ **D7.** *Fraud Flagging of Credentials.* S allows authorized parties (e.g. U and I) to flag x_U as fraudulent (i.e. indicate in a trusted accessible database that, for a specified period, all uses of x_U* are fraudulent). Evaluation: When U suspects that any of her cred-tokens or any cred-info thereon is stolen or cloned, U notifies F accordingly; then, F declares each transaction involving U's cred-info as fraudulent, until F and I complete the fraud recovery procedure (see the Fraud Recovery protocol in §2.3).

✓ **D8.** *Verification of Credential Fraud Flag.* R can know whether x_U* is currently flagged as fraudulent. Evaluation: Since, at each transaction, R receives a fraud status notification from F, R can reject or approve of each transaction based on up-to-date fraud status information; (the faster messages are communicated between F and I, and I and R, the more up-to-date the information received by R is; If D_U, R, I, and F have synchronized clocks, then these parties may even include a time stamp in each message they encrypt or sign. Among other things, this would allow R to determine whether fraud status notifications it receives are outdated (using a predefined expiration rule)).

✓ **D9.***Detection of Clone Usage.* R (resp. I) can distinguish† x_U from its clones whenever the latter are presented to R (resp. I). Evaluation: The need for an attacker to provide F with a correct $k^{(i)}$ helps F distinguish the presentation of C_U by U from an analogous presentation by an attacher A. Hence, S detects the fraudulent use of cred-tokens or cred-info.

✗ **D10.** *Detection of Credential Cloning.* U and I can detect† that x_U* is cloned. Evaluation: S does not detect the actual theft or cloning of cred-tokens or cred-info.

✗ **D11.** *Detection of Credential Theft.* U and I can detect† that x_U is stolen from U. Evaluation: See `D10`.

✓ **D12.** *Determination of Malicious Fraud Claims.* I can determine† whether U is honest when claiming that x_U* has been used without proper authorization. Evaluation: S does not detect instances of fraud that are intentionally committed by U.

Communication Security Evaluation Criteria

✓ **C1.** *Protection Against Physical Exposure of x_U*.* S protects x_U* from being visually captured (e.g. via shoulder surfing) by unauthorized parties, without requiring U's conscious cooperation. Evaluation: The use of a small portable device D_U is partially intended to reduce the risk of shoulder surfing attacks when U enters p_U in D_U.

✓ **C2.** *Protection Against Digital Exposure of x_U*.* If x_U* is cred-info, S protects x_U* from being accessed by unauthorized parties using computer systems. For example, S may protect x_U* from being captured in an intelligible form when x_U* is communicated over untrusted channels (e.g. the Internet). Evaluation:

[11] Note that this criterion may adversely affect user privacy.

[12] For example, x_U*'s holder may claim to be *Joe Dalton* while x_U* was issued to *Lucky Luke*. This is different from the situation in which x_U's holder pretends to be Lucky Luke (see D3).

In S, x_U^* is not sent in the clear: a MAC thereof is computed with a high-entropy key, and this MACed value is sent. The secrecy and entropy of the key protects x_U^* from being derived from the MACed value.

✓ **C3.** *Protection Against Replay Attacks.* S prevents (or reduces to a negligible proportion) reuse of electronic messages sent to impersonate U. Evaluation: S uses communication channels providing message-replay detection for a chosen time frame.

✓ **C4.** *Protection Against Man-In-The-Middle Attacks.* S prevents (or reduces to a negligible proportion) impersonation of U through tampering or injection of messages between parties used by S. Evaluation: Assuming the MAC algorithm used to generate them is secure, the tokens sent by D_U are integrity-protected. Hence, S provides Man-In-The-Middle protection. In addition, S uses channels providing integrity protection between R and I, and I and F.

✗ **C5.** *Protection Against Denial of Service Attacks.* S prevents (or reduces to a negligible proportion) denial of services against U. Evaluation: Provided an attacker A knows ID_U, A can generate and send to F incorrect cred-tokens. This would cause a denial of service against U (after a small number of failed transactions, all uses of U' cred-tokens will be temporarily considered as fraudulent).

Specific applications may require that subsets of the proposed criteria be met (as best as possible), but universal IDF solutions may be required to meet many or even all criteria. For practical purposes, instant detection of credential cloning and theft (see D10 and D11) might be optional for universal IDF solutions; the existence of cloned cred-tokens may be more difficult to detect (with current technologies) than their use.

Based on the above criteria, `CROO` is expected[13] to provide usability benefits for both users and relying parties; to detect IDF attempts; to identify cases of committed yet previously undetected IDF; and to be resistant to a number of communication-based attacks (e.g. replay and man-in-the-middle attacks, including phishing and PC-based key logging). Two limitations of `CROO` are: its inability to detect cases in which legitimate users perform transactions, and later repudiate them; and susceptibility to denial of service attacks against specific users, by attackers who have gathered sufficient and correct credential information.

3.2 Relative Importance of Security and Privacy-Related Design Requirements

We now discuss some security and privacy-related design requirements of `CROO`.

1. *Trusted User Devices*: `CROO` makes use of trusted user devices. Many widely-deployed PDAs and cell phones are arguably more trustworthy than commonplace home PCs, even though this might change in the future [33]. For increased user device trustworthiness, one can use formally verifiable OS kernels [59], or trusted virtual machine monitors [20], among other alternatives [2]. While implementing `CROO` with *any* existing cell phones or PDAs might not be appropriate, we believe that some (more trusted) smartphones (e.g. Blackberry 8700 series) could provide a sufficiently trustworthy computing platform for applications such as credit or debit card transactions.
2. *User Devices Performing Cryptographic Operations*: `CROO` relies on user devices performing common cryptographic computations (e.g. hashing). These operations are provided by the Bouncy Castle Crypto API for CLDC/MIDP-enabled devices. MIDP 1.1 is supported by a large proportion of widely-deployed cell phones and PDAs.
3. *One-Time (vs. Static) Passwords*: `CROO` uses OTPs instead of static passwords (shared secrets), because the latter do not allow to detect lack of transaction-counter synchrony between user devices and their associated fraud detecting parties (F).
4. *Identification of Previously Undetected IDF*: One may distinguish the detection of IDF attempts, from the identification of previously undetected IDF. The former primarily aims at preventing IDF from happening; the latter primarily aims at limiting the damages of IDF that has occurred. `CROO` provides means to detect IDF attempts, and, when these fails (e.g. due to more sophisticated attacks), provides means to identify previously undetected IDF.
5. *Transaction Approval by Credential Issuers*: The protocol presented in §2.3 requires I to approve each transaction associated with credential information issued by I. This may be suitable for applications such as credit or debit card transactions, but not for other applications (e.g. due to privacy concerns). For the latter, the protocol presented in §2.3 might be adapted.
6. *Disclosure of Transaction Details to Credential Issuers*: In the §2.3 protocol, I receives transaction details from R. To protect user privacy, care must therefore be taken to reveal only information required to authorize transactions (e.g. transaction date, time, and dollar value).

[13] This design-level paper considers a number of theoretical and practical issues of IDF detection. We have not empirically confirmed our usability analysis through a prototype implementation, user lab, or field tests. This is left for future work.

7. *Detection of Replay Attacks by Credential Relying Parties*: The protocol of §2.3 requires R to detect replay attacks (for a chosen time frame). While this might be seen as proper input validation on R's part, the requirement could be removed.
8. *Logical Distinction between Parties Involved in Protocol*: The protocol presented in §2.3 logically distinguishes U, R, I, and F based on their respective roles. However, these parties may be collocated or incarnated by single entities in concrete instantiations, e.g. with $I = F$ (as in the Student IDF scenario presented in §2.4), or $I = R$. When $R = I$ (resp. $I = F$), the protocol steps ensuring secure communication between R and I (resp. I and F) can be removed. While we do not find concrete applications with $R = F \neq I$, the analysis of this scenario is left for future work.
9. *Online Fraud Detection*: `CROO` is an online protocol. To *prevent* IDF using an offline protocol is a challenging task. Offline IDF prevention employing user-specific cryptographic keys generated by personal user devices seems to require user devices that are tightly bound to their legitimate users (i.e. which, in practice, cannot be used by impersonators). Tight user-device binding may be achieved through authentication of users via practically unforgeable biometric techniques. Devices providing such a guarantee are currently not widely deployed, and it is not clear if or when they will be. Consequently, we favor online fraud detection, and note that recent trends in computing and communications are towards online protocols (e.g. a significant proportion of widely deployed cell phones are able to connect to the Internet).

3.3 Preliminary Mathematical Security Analysis

Having analyzed `CROO` from several practical standpoints, we now present a brief mathematical security analysis of the proposed scheme. To do so, we first model `CROO` as a simplified scheme (henceforth called Σ) meant to capture the secrecy and forgery resistance aspects of `CROO`. Our analysis does not focus on easier-to-prove features of `CROO`, e.g. completeness (U can always generate valid user tokens (ID_U, v)) and soundness (given $h(z)$, F can always verify a user token (ID_U, v)). Secrecy and forgery protection are tied with the use of secrets in `CROO`. Soundness and completeness are tied with the use of OTPs as a particular class of secrets. Hence, Σ does not capture the use of OTPs in `CROO`. Here is Σ:

- `Setup` (w): Given a security parameter w as input, this algorithm outputs a tuple (d, s_U, h, f, G), where $d = (d_1, d_2, d_3, d_4, d_5)$ and $G : \{0,1\}^{d_2} \rightarrow \{0,1\}^{d_5}$ is such that $G(x) = x \ mod \ 2^{d_5}$, f is a family of functions that can be modeled as a pseudorandom function (PRF), and the other symbols are defined as in Table 1 with the extra assumption that d_1 through d_5 are polynomial values in w.
- `Gen-Cred` (z, s_U, h, f): Given (z, s_U, h, f), this algorithm outputs $v = G(f_k(z))$, where $k = h(s_U)$.
- `Verify-Cred` (z, s_U, h, f, v): Given (z, s_U, h, f, v), this algorithm outputs `True` if $v = G(f_k(z))$, where $k = h(s_U)$. Otherwise, this algorithm returns `False`.

We consider the following *Secrecy* game between a challenger C and an attacker A:

- `Initialization` (w): Given w, C runs `Setup`(w), gets (d, s_U, h, f, G), and sends (d, h, f, G) to A.
- `Queries`: Let N be a polynomial value in w generated by A. A sends N (adaptive) queries z_i to C, and, for each query, C replies with a value v_i such that `Verify-Cred`(z_i, s_U, h, f, v_i) =`True`.
- `Submission`: A sends $s'_U \in \{0,1\}^{d_2}$ to C, and wins the game if, for all $i = 1, \cdots, N$, the following condition is met: `Verify-Cred`(z_i, s'_U, h, f, v_i) =`True`.

The probability that A wins the *Secrecy* game is called the advantage of A in the game. Σ is said to provide *secrecy* if no polynomial attacker has non-negligible advantage in the *Secrecy* game. We also consider the following *Forgery* game between a challenger C and an attacker A:

- `Initialization` (w): Given a security parameter w, C runs `Setup`(w), obtains (d, s_U, h, f, G), and sends (d, h, f, G) to A.
- `Queries`: Let N be a polynomial value in w generated by A. A sends N (adaptive) queries z_i to C, and, for each query, C replies with a value v_i such that `Verify-Cred`(z_i, s_U, h, f, v_i) =`True`.
- `Submission`: A sends to C a pair (z, v) such that $z \notin \{z_1, \cdots, z_N\}$, and wins the game if and only if `Verify-Cred`(z, s_U, h, f, v) =`True`

The probability that A wins the *Forgery* game is called the advantage of A in the game. Σ is said to provide *forgery resistance* if no polynomial attacker has non-negligible advantage in the *Forgery* game.

Theorem 1 Σ *provides* secrecy.

Proof: To win the *Secrecy* game, any polynomial attacher A only has the following options: (a) A randomly guesses $s_U \in \{0,1\}^{d_2}$; (b) A randomly guesses k and reverses h to find s_U; (c) given some v_i sent to A in the `Queries` phase, A reverses $G \circ f$ to find k, and then reverses h to find s_U. The probably that A wins in case (a) is less that $\frac{1}{2^{d_2}} + \nu(w)$, where ν is a negligible function. The probability that A wins in case (b) is less than $\frac{1}{2^{d_2}} + \nu(w)$, since h is a one-way function. Hence, the probability that A wins in case (c) is less than $\nu(w) + (\frac{1}{2^{d_2}} + \nu(w))$ since f can be modeled as a PRF and this implies (see, e.g., proof of Theorem 1 in [46]) that $G \circ f$ can be modeled as a PRF. Consequently, the advantage of A in the game is less than $\frac{3}{2^{d_2}} + 4\nu(w)$. Since d_2 is polynomial in w, we conclude that A's advantage in the secrecy game is negligible. QED.

Theorem 2 Σ *provides* forgery resistance.

Proof: To win the *Forgery* game, any polynomial attacker A only has the following options: (a) A picks any $z \in \{0,1\}^*$, randomly guesses $v \in \{0,1\}^{d_5}$ hoping `Verify-Cred`$(z, s_U, h, f, v) =$`True`, and submits (z, v) to C; (b) A randomly guesses $s_U \in \{0,1\}^{d_2}$, picks any $z \in \{0,1\}^*$, and submits (z, v) to C where $v = G(f_k(z))$; (c) A randomly guesses $k \in^{d_2}$, picks any $z \in \{0,1\}^*$, and submits (z, v) to C where $v = G(f_k(z))$; (d) for any value v_i sent by C in the `Queries` phase, A reverses $G \circ f$ to find k, then picks any $z \in \{0,1\}^*$ and submits (z, v) to C such that $v = G(f_k(z))$; (e) A picks any $v \in \{0,1\}^{d_5}$, randomly guesses $z \in \{0,1\}^{d_5}$ hoping `Verify-Cred`$(z, s_U, h, f, v) =$`True`, and submits (z, v) to C. The probability that A wins in case (a) is less than $\frac{1}{2^{d_5}} + \nu(w)$, where ν is a negligible function. The probability that A wins in case (b) is less than $\frac{1}{2^{d_2}} + n(w)$. The probability that A wins in case (c) is also less than $\frac{1}{2^{d_2}} + n(w)$. The probability that A wins in case (d), is less than $\frac{1}{2^{d_2}} + n(w)$ since $G \circ f$ can be modeled as a PRF. The probability that A wins in case (e) is less than νw since the size of z can be arbitrary. Hence, the advantage of A in the *Forgery* game is less than $\frac{5}{2^{min\{d_5,d_2\}}} + 5\nu(w)$. Both d_2 and d_5 are polynomial values in w. Consequently, the advantage of A in the *forgery* game is negligible in w. QED.

Discussion The above analysis shows that the secrecy (respectively the forgery resistance) of Σ (and by extension `CROO`) is limited by d_2 (respectively by d_5). Since, in practice d_5 is constrained by the entropy of well-formatted user credentials, one may say that, in practice, Σ might provide more secrecy than forgery resistance, even though, in practice forgery attempts are very likely to be detected (and thwarted) before successful forgery occurs.

4 Related Work and Comparison

The design of `CROO` involves consideration for various aspects including: IDF detection (before and after fraud); limitation of IDF consequences; methodological generality (universality); device capture resilience; and deployability. In the following paragraphs, we review work related to these aspects (including research on specific applications, e.g. electronic payment).

Password-based Authentication. Static password schemes,[14] one-time password (OTP) schemes [35], password schemes resilient to shoulder surfing attacks [25, 36, 53, 60], and schemes generating domain-specific passwords from a combination of single user-chosen passwords and multiple domain-specific keys [52, 24] can all be used to authenticate users and thereby solve parts of the problem of phishing and/or IDF. System designers must carefully select and combine multiple cryptographic and non-cryptographic tools and techniques to meet various requirements, e.g. those presented in §3.1. Our scheme can be viewed as a careful combination of known and modified tools and techniques (e.g. cell phones, non-verifiable text [37], OTP-based authentication, and symmetric and asymmetric cryptography) to detect IDF.

Limited-Use Credit Card Numbers. Rubin and Wright [54] propose a scheme for off-line generation of limited-use (e.g. one-time) credit card (CC) numbers. While similar in some ways, `CROO` is universal, and is designed to counter device capture attacks (through the use of PIN-encrypted unverifiable keys).[15] Singh and dos Santos [56] describe another scheme for off-line generation of limited-use credentials. Unlike our scheme, Singh and dos Santos' is not meant to be universal and counter device capture attacks. Shamir [55] describes a scheme to generate one-time CC numbers via an online interactive procedure whereby CC holders obtain these numbers from CC issuers. The number-generation procedure in Shamir's scheme can be automated using

[14] Including commonplace typed textual password mechanisms, and strengthened password schemes [1, 24, 42].

[15] Note that limited-use CC numbers may follow the format of existing CC numbers. This provides the benefit of requiring no change in existing CC number validation software.

a plugin installed on user PCs. This does not (and is not meant to) counter attacks whereby users' browsers or PCs are compromised e.g. via PC-based virus infection or key-logging attacks. Molloy et al. [46] propose a scheme for off-line generation of limited-use credit card numbers; their scheme is susceptible to dictionary attacks on user passwords.

Smartcard-based Authentication. Lu and Ali [38] propose the use of network smart cards to secure web-based transactions. As presented, this scheme does not identify cases of committed yet previously undetected IDF. A compromised PC equipped with a key logger may also maliciously exploit a user-input PIN to control the user's smartcard, when this card is in the PC's smartcard reader.

Detection of Compromised Cryptographic Signature Keys. Just and van Oorschot [28] describe a scheme to detect fraudulent client-based cryptographic signatures. We deal with the more general context of fraudulent uses of credential information.

Limiting the Effect of Cryptographic Key Exposure. Public-key schemes have been proposed to limit the effect of key exposure. These schemes include threshold cryptosystems [8], proactive cryptosystems [26], proactive forward-secure schemes [4], key-insulated cryptosystems [15], and intrusion-resilient cryptosystems [14]. All are designed to decrease the odds that unauthorized public-key signatures be issued; we deal with the general problem of IDF committed with cloned credential information.

Device Capture Resilience. The idea of capture resilience was suggested by Mackenzie et al. [40, 41] to detect attempts of off-line password-guessing attacks on password-protected mobile devices, by requiring password-based user-to-device authentication to be mediated (and, *ergo*, detectable) by online servers. This mediation is implemented using the concepts of partial decryption; a similar proposal by Tang et al. [58] is implemented using static PINs partially stored on mobile devices and on an online server. Our proposal is implemented using one-time passwords generated from non-verifiable PIN-protected text stored on mobile devices. In addition, for easier deployability, `CROO` generates user credentials which can be formatted as existing, typically low-entropy credentials (e.g. credit card numbers), while the aforementioned capture-resilient schemes use either high-entropy cryptographic keys or public-key encrypted PINs as user credentials.

OTP-Generating Tokens. Various companies (e.g. Aladdin, RSA and Mastercard) have developed variants of a scheme whereby hardware tokens or mobile device software are used to generate OTPs which are then manually input into PCs, in cleartext form, for remote user authentication and/or transaction authorization. Existing variants of this scheme are not (and not meant to be) simultaneously universal, usable without a vendor-specific hardware token, resilient to device capture, and immune to phishing and PC-based key-logging attacks whereby OTPs are copied and then used for unintended transactions.

IDF Detection via Location Corroboration. Van Oorschot and Stubblebine [61] propose an IDF detection scheme, whereby users' identity claims are corroborated with trusted claims of these users' location. The scheme requires users to be geographically tracked; this may be a concern for user location privacy.[16] A benefit of the scheme is that it can be used regardless of transactions' purpose and associated credentials. It is designed for on-site transactions. Mannan and van Oorschot [43] propose an authentication protocol involving an independent personal device, and survey related schemes.

SET and Certificate-Based PKIs. The Secure Electronic Transaction (SET) protocol [23] allows credit card (CC) holders to obtain goods or services from merchants without revealing their CC information to the latter. SET is not designed to be used for multiple classes of cred-tokens, nor does it specify methods to identify cases of committed yet undetected IDF. SET also employs user-specific (i.e. CC holder) private keys in a certificate-based public-key infrastructure (PKI); we favor the use of OTPs as user authentication secrets, mostly because their misuse can be subsequently detected and their misuse detection does not call for an associated notification to a potentially large population of parties relying on the validity of public-key certificates associated with compromised signing keys. SET also uses high-entropy user-keys, whereas, for easier deployability, `CROO` allows to format user-credentials as existing low-entropy credentials.

Biometric-based Authentication. Kwon and Moon [32] discuss the use of multi-modal biometric authentication as a way to improve unimodal biometric user identification, and thereby decrease the risk of associated forms of IDF (e.g. immigration fraud). Biometric-based authentication has known limitations. (See, e.g., [11], Chap. 10, p. 104; in practice, it is difficult to instantiate systems using trusted biometric readers intended to be deployed on remote trusted user PCs.)

Web Authentication through Manual Input of OTPs Sent to Mobile Phone via SMS. Signify is a company providing a service, whereby each user gives a mobile phone number, userid, and password to an

[16] One may argue that if `CROO` is instantiated with mobile phones as user devices, `CROO` is also susceptible to user location tracking by mobile phone call carriers.

authentication server (auth-server), at registration time. To access restricted services or resources over the web, each user must manually input her userid and password into a PC, send these two pieces of data to a web-based auth-server, receive an OTP via SMS (at the phone number specified at registration), manually input this OTP into the PC, and send the OTP over the web to the auth-server. If the web-based auth-server receives the OTP that was sent to a user's mobile phone, this user is authenticated. This scheme is not (and not meant to be) resilient to PC-based key-logging attacks whereby userids, passwords and OTPs are captured and used without proper authorization.

Mobile Payment Schemes. MobileLime [45] is a payment system in which nomadic users are enrolled via a web interface. To be enrolled, each user provides her name and mobile phone number, and associates this information with a prepaid or credit card account. To pay for an item or service at an authorized point of sale (POS), a MobileLime user dials a toll free number using her cell phone, and enters both a PIN (provided at enrollment) and a number identifying the POS. When a payment is completed, a text receipt is sent to the associated user's mobile phone; this can later be used as proof of payment. Other mobile payment systems include Black Lab Mobile's system [6], Sonera's Shopper [16], and Parkit [27]. Some payment systems require physical proximity of POS terminals and buyers' mobile devices. These include Ravi et al.'s system [50], and sytems marketed by PayCircle [49], Encorus PaymentWorks Mobile [17], and Ilium Software [57]. None of these mobile payment systems are designed to identify previously undetected fraudulent financial payments (e.g. when users' mobile phones are stolen and misused for unauthorized CC transactions).

Web Authentication from Untrusted Terminals. Oprea et al. [48] propose a scheme whereby a remote application is accessed through a mobile trusted device. This mobile device (e.g. PDA) delegates to an untrusted terminal temporary access to the remote application. More precisely, the mobile device establishes an SSL session with the remote application and maintains a secure (confidential and authenticated) communication link with the untrusted terminal. The remote application also establishes an associated SSL session with the untrusted terminal. The user issues requests to the remote application via the trusted mobile device, and the remote application sends encrypted answers to these requests to the untrusted terminal. The mobile trusted device provides temporary decryption keys to the untrusted terminal, thereby enabling this terminal to access portions of the encrypted information sent by the remote application for the user to view on the untrusted terminal. This scheme grants to the untrusted terminal temporary access to confidential information. The scheme is not meant to detect misuse of stolen trusted mobile devices or cryptographic keys stored therein. Clarke et al. [10] propose a scheme enabling bidirectional authenticated communication between a user and an associated trusted proxy via an untrusted terminal. The trusted proxy may be granted the authority to act on behalf of the user (e.g. to utilize the user's passwords to establish SSL connections with trusted web sites). This scheme relies on camera-enabled trusted mobile devices carried by users, and works as follows. The trusted mobile device video-captures the information displayed by the untrusted terminal, and compares it with confidential information received from the trusted proxy via an untrusted communication channel. (This communication channel is used by both the untrusted terminal and the mobile device in order to communicate with the proxy.) When the information displayed by the untrusted terminal does not match the value expected by the mobile device, the user is notified. Clarke et al. [10] specify that access to the trusted mobile devices should be protected by PINs or biometrics, thereby partially addressing the possibility of mobile device theft. The scheme is not designed to identify cases in which users' mobile devices have been used to impersonate users without proper authorization. Wu et al. [62] described a scheme enabling a user U to access a remote web-server R via an untrusted kiosk K. U uses a trusted proxy P to mediate all communication with R and utilizes a mobile phone M to communicate with P via SMS. P keeps U's passwords and can securely communicate with R. The scheme works as follows: (a) U directs K to P, specifying both her username and R's identifier (e.g. web address); (b) P sends a session name to K (via the web) and sends the same session name to M via SMS; (c) If U finds that the two session names match, she allows[17] a secure session to be established between P and R. The scheme is not designed to counter mobile phone theft or cloning. Balfanz and Felten [3] propose a method to prevent misuse or disclosure of signing and decryption cryptographic keys stored on smartcards. The idea is to use a trusted PDA (instead of a smartcard) to request from its user the authorization to perform cryptographic operations with secret keys stored on the PDA. The scheme is not designed to detect misuse of secret keys stored on stolen or cloned PDAs.

5 Concluding Remarks

We address the general problem of IDF. We propose criteria to characterize and compare instances of IDF, providing a framework to evaluate IDF solutions by examining the usability, privacy-preserving capability, fraud detection capability, and communication security of these solutions. We argue that complete IDF solutions

[17] This is done by pressing a button on M and thereby sending a web message to P.

should provide mechanisms that detect the use of compromised private credential information. Our proposed scheme (CROO) implements this idea without requiring the collection of private behavioral information (in contrast to statistical anomaly-based fraud detection schemes used, e.g., by banks to detect credit card fraud). CROO associates each use of credential information with a one-time password verified by an online trusted party F. F need not be the same for all users (thus improving scalability). An important feature of CROO is its universal nature, i.e. it is designed to simultaneously be used with multiple classes of applications and credential tokens, in both online and on-site transactions. CROO's user credentials can be formatted as existing user credentials, thereby making potentially easier the adoption of the proposed scheme. CROO also allows each IDF victim to continue to use her credential tokens (e.g. credit cards) provided she uses her portable trusted device to send new one-time password setup information to F. This feature can be useful when it is preferable (e.g. for time efficiency, convenience, or lack of alternative options) to continue to use credential tokens, even though they have been cloned, rather than obtaining new ones. This is appealing in cases in which it takes less time to go in person to a single local party F (e.g. a trusted government agency's office) to give new OTP setup information, than having social security numbers replaced, or obtaining new credit cards by postal mail. We encourage work on mathematical models that help evaluate IDF detection schemes, but note the challenge of generating *realistic* models (particularly for universal schemes). We also encourage further exploration in the design of schemes that detect fraudulent uses of compromised authentication keys.

Acknowledgement. We thank M. Mannan for helpful discussions, and anonymous referees for comments improving both technical and editorial aspects of the paper. We acknowledge partial funding from ORNEC, and the second author acknowledges NSERC for funding an NSERC Discovery Grant and his Canada Research Chair in Network and Software Security.

References

[1] M. Abadi, T.M.A. Lomas, and R. Needham. Strengthening Passwords. Technical Report 1997 - 033, Digital Equipment Corporation, 1997.

[2] B. Balacheff, L. Chen, S. Pearson, D. Plaquin, and G. Proudler. *Trusted Computing Platforms TCPA Technology in Context*. Prentice Hall, 2003.

[3] D. Balfanz and E. Felten. Hand-Held Computers Can Be Better Smart Cards. In *USENIX Security Symposium*, Washington, DC, August 1999.

[4] M. Bellare and S.K. Miner. A Forward-Secure Digital Signature Scheme. In *CRYPTO '99*, volume 1666 of *Lecture Notes in Computer Science*, pages 431–448. Springer-Verlag, 1999.

[5] Better Busineess Bureau (USA). Victims' Stories. `http://www.bbbonline.org/idtheft/stories.asp`. Site accessed in Jan. 2008.

[6] Black Lab Mobile. `http://www.blacklabmobile.com/`. Site accessed in Jan. 2008.

[7] S. Brands. *Rethinking Public Key Infrastructures and Digital Certificates; Building in Privacy*. MIT Press, 2000.

[8] R. Canetti and S. Goldwasser. An Efficient Threshold Public-Key Cryptosystem Secure Against Adaptive Chosen Ciphertext Attack. In *EUROCRYPT '99*, volume 1592 of *Lecture Notes in Computer Science*, pages 90–106. Springer-Verlag, 1999.

[9] N. Chou, R. Ledesma, Y. Teraguchi, D. Boneh, and J.C. Mitchell. Client-Side Defense Against Web-Based Identity Theft. In *Network and Distributed System Security Symposium (NDSS '04)*. The Internet Society, 2004.

[10] D.E. Clarke, B. Gassend, T. Kotwal, M. Burnside, M. van Dijk, S. Devadas, and R.L. Rivest. The Untrusted Computer Problem and Camera-Based Authentication. In *International Conference on Pervasive Computing*, volume 2414 of *LNCS*, pages 114–124. Springer-Verlag, 2002.

[11] L. Cranor and S. Garfinkel. *Security and Usability*. O'Reilly Media, Inc., August 2005.

[12] DataTex Engineering Corporation. Computer Crime, Cyber Crime, and Identity Theft Are you a Victim? `http://www.datatexcorp.com/html/cybercrimenews.htm`. Site accessed in Jan. 2008.

[13] R. Dhamija and J. D. Tygar. The Battle Against Phishing: Dynamic Security Skins. In *Symposium on Usable Privacy and Security (SOUPS '05)*, pages 77–88. ACM Press, 2005.

[14] Y. Dodis, M. Franklin, J. Katz, A. Miyaji, and M. Yung. Key-Insulated Public Key Cryptosystems. In *CT-RSA '03*, volume 2612 of *Lecture Notes in Computer Science*, pages 19–32. Springer-Verlag, 2003.

[15] Y. Dodis, J. Katz, S. Xu, and M. Yung. Key-Insulated Public Key Cryptosystems. In *EUROCRYPT '02*, volume 2332 of *Lecture Notes in Computer Science*, pages 65–82. Springer-Verlag, 2002.

[16] eFinland. Mobile Services. `http://e.finland.fi/netcomm/news/showarticle.asp?intNWSAID=7240`. Site accessed in Jan. 2008.

[17] Encorus. PaymentWorks Mobile. `http://www.wi-fitechnology.com/printarticle535.html`. Site accessed in Jan. 2008.

[18] Australian Center for Policing Research. Standardization of Definitions of Identity Crime Terms - Discussion Paper, Prepared by the Australian Center for Policing Research for the Police Commissioners' Australian Identity Crime Working Party and the AUSTRAC POI Steering Committee, 2005.

[19] NFC Forum. `http://www.nfc-forum.org/home`. Site accessed in Jan. 2008.

[20] T. Garfinkel, B. Pfaff, J. Chow, M. Rosenblum, and D. Boneh. Terra: A virtual machine-based platform for trusted computing. In *ACM Symposium on Operating Systems Principles (SOSP'03)*, 2003.

[21] M.T. Goodrich, M. Sirivianos, J. Solis, G. Tsudik, and E. Uzun. Loud and Clear: Human-Verifiable Authentication Based on Audio. In *IEEE International Conference on Distributed Computing Systems (ICDCS' 06)*. IEEE, 2006.

[22] G.R. Gordon and N.A. Willox. Identity Fraud: A Critical National and Global Threat. *Journal of Economic Crime Management*, 2(1):1–47, 2005.

[23] Network Working Group. RFC 3538 - Secure Electronic Transaction (SET) Supplement for the v1.0 Internet Open Trading Protocol (IOTP), 2003. `http://www.faqs.org/rfcs/rfc3538.html`. Site accessed in Jan. 2008.

[24] J.A. Halderman, B. Waters, and E.W. Felten. A Convenient Method for Securely Managing Passwords. In *International Conference on World Wide Web (WWW '05)*, pages 471–479. ACM Press, 2005.

[25] J.A. Haskett. Pass-algorithms: A User Validation Scheme Based on Knowledge of Secret Algorithm. *Communications of the ACM*, 27(8):777–781, 1984.

[26] A. Herzberg, M. Jakobsson, S. Jarecki, H. Krawczyk, and M. Yung. Proactive Public Key and Signature Systems. In *ACM Conference on Computer and Communications Security (CCS '97)*, pages 100–110. ACM Press, 1997.

[27] PARK IT. `http://www.payway.fi/`. Site accessed in Jan. 2008.

[28] M. Just and P.C. van Oorschot. Addressing the Problem of Undetected Signature Key Compromise. In *Network and Distributed System Security (NDSS '99)*. The Internet Society, 1999.

[29] E. Kirda and C. Kruegel. Protecting Users Against Phishing Attacks with AntiPhish. In *Computer Software and Applications Conference '05*, pages 517–524, 2005.

[30] N. Koblitz and A.J. Menezes. Another look at provable security II. In *INDOCRYPT 2006*, volume 4329 of *Lecture Notes in Computer Science*, pages 148–175. Springer-Verlag, 2006.

[31] N. Koblitz and A.J. Menezes. Another look at provable security. *Journal of Cryptology*, 20(1):3–37, 2007.

[32] T. Kwon and H. Moon. Multi-modal Techniques for Identity Theft Prevention. In *International Conference on Human.Society@Internet*, volume 3597 of *Lecture Notes in Computer Science*, pages 291–300. Springer-Verlag, 2005.

[33] Kaspersky Lab. Trojan targets mobile phones running Java applications., 2006. `http://www.kaspersky.com/news?id=180984542`. Site accessed in Jan. 2008.

[34] D. Lacey and S. Cuganesan. The Role of Organizations in Identity Theft Response: the Organization-Individual Dynamic. *Journal of Consumer Affairs*, 38(2):244–261, 2004.

[35] L. Lamport. Password Authentication with Insecure Communication. *Communications of the ACM*, 24:770–772, 1981.

[36] S. Li and H.-Y. Shum. Secure Human-Computer Identification (Interface) Systems against Peeping Attacks. Cryptology ePrint Archive, Report 2005/268, 2005.

[37] T.M.A. Lomas, L. Gong, J.H. Saltzer, and R.M. Needham. Reducing Risks from Poorly Chosen Keys. *ACM SIGOPS Operating Systems Review*, 23(5), 1989.

[38] H.K. Lu and A. Ali. Prevent Online Identity Theft Using Network Smart for Secure Online Transactions. In *International Conference on Information Security (ISC '04)*, volume 3225 of *Lecture Notes in Computer Science*, pages 342–353. Springer-Verlag, 2004.

[39] M. Jakobsson. Modeling and Preventing Phishing Attacks, 2005. Phishing Panel in Financial Cryptography (FC '05).

[40] P. MacKenzie and M.K. Reiter. Networked Cryptographic Devices Resilient to Capture. In *IEEE Symposium on Security and Privacy*, pages 12–25. IEEE Computer Society, 2001.

[41] P. MacKenzie and M.K. Reiter. Delegation of cryptographic servers for capture-resilient devices. *Distributed Computing*, 16(4):307–327, 2003.

[42] U. Manber. A Simple Scheme to Make Passwords Based on One-Way Functions Much Harder to Crack. *Computers and Security*, (2):171–176, 1996.

[43] M. Mannan and P.C. van Oorschot. Using a personal device to strengthen password authentication from an untrusted computer. In *Financial Cryptography 2007 (FC '07)*, volume 4886 of *Lecture Notes in Computer Science*, pages 88–103. Springer-Verlag, 2007.

[44] J.M. McCune, A. Perrig, and M.K. Reiter. Seeing-is-believing: Using camera phones for human-verifiable authentication. In *IEEE Symposium on Security and Privacy*, May 2005.

[45] MobileLime. `https://www.mobilelime.com/mobilelime/home.do?action=index`. Site accessed in Jan. 2008.

[46] I. Molloy, J. Li, and N. Li. Dynamic virtual credit card numbers. In *Financial Cryptography 2007 (FC '07)*, volume 4886 of *Lecture Notes in Computer Science*, pages 208–223. Springer-Verlag, 2007.

[47] D. Nali and P.C. van Oorschot. CROO: A Generic Architecture and Protocol to Detect Identity Fraud. Technical Report, TR-06-14, School of Computer Science, Carleton University, Ottawa, Canada, Dec. 20, 2006.

[48] A. Oprea, D. Balfanz, G. Durfee, and D. Smetters. Securing a Remote Terminal Application with a Mobile Trusted Device. In *Annual Computer Security Applications Conference (ACSAC '04)*, Phoenix, AZ, December 2004.

[49] PayCircle. `http://www.paycircle.org/`. Site accessed in Jan. 2008.

[50] N. Ravi, P. Stern, N. Desai, and L. Iftode. Accessing Ubiquitous Services Using Smart Phones. In *International Conference on Pervasive Computing and Communications (PerCom 2005)*, 2005.

[51] Javelin Strategy & Research. 2005 Identity Fraud Survey Report, 2005. `http://www.javelinstrategy.com/reports/2005IdentityFraudSurveyReport.html`.

[52] B. Ross, C. Jackson, N. Miyake, D. Boneh, and J.C. Mitchell. Stronger Password Authentication Using Browser Extensions. In *USENIX Security Symposium*, pages 17–32, 2005.

[53] V. Roth, K. Richter, and R. Freidinger. A PIN-Entry Method Resilient Against Shoulder Surfing. In *ACM Conference on Computer and Communications Security (CCS '04)*, pages 236–245. ACM Press, 2004.

[54] A.D. Rubin and R.N. Wright. Off-line generation of limited-use credit card numbers. In *Financial Cryptography 2001 (FC '01)*, volume 2339 of *Lecture Notes in Computer Science*, pages 196–209. Springer-Verlag, 2002.

[55] A. Shamir. Secureclick: A web payment system with disposable credit card numbers. In *Financial Cryptography 2001 (FC '01)*, volume 2339 of *Lecture Notes in Computer Science*, pages 232–242. Springer-Verlag, 2002.

[56] A. Singh and A.L.M. dos Santos. Grammar based off line generation of disposable credit card numbers. In *ACM Symposium on Applied Computing 2002 (SAC '02)*, pages 221–228. ACM Press, 2002.

[57] Ilium Software. eWallet. `http://www.iliumsoft.com/site/ew/ewallet.htm`. Site accessed in Jan. 2008.

[58] J. Tang, V. Terziyan, and J. Veijalainen. Distributed PIN Verification Scheme for Improving Security of Mobile Devices. *Journal of Mobile and Network Application*, 8(2):159–175, 2003.

[59] H. Tuch, G. Klein, and G. Heiser. OS verification - now! In *Workshop on Hot Topics in Operating Systems (HotOS X)*, 2005.

[60] T. Valentine and M. Endo. Towards an Exemplar Model of Face Processing: the Effects of Race and Distinctiveness. *Quaterly Journal of Experimental Psychology*, 44:671–703, 1992.

[61] P.C. van Oorschot and S. Stubblebine. Countering Identity Theft through Digital Uniqueness, Location Cross-Checking, and Funneling. In *Financial Cryptography and Data Security 2005 (FC '05)*, volume 3570 of *Lecture Notes in Computer Science*, pages 31–43. Springer-Verlag, 2005.

[62] M. Wu, S. Garfinkel, and R. Miller. Secure Web Authentication with Mobile Phones. In *DIMACS Workshop on Usable Privacy and Security Systems*, 2004.