CROO: A Universal Infrastructure and Protocol to Detect Identity Fraud

D. Nali and P.C. van Oorschot

School of Computer Science, Carleton University, Ottawa, Canada

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 can be used regardless of a 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' devices resilient to off-line PIN-guessing attacks. CROO's credentials can be formatted 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 [25] 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 (including, e.g., loss of privacy, worry and fear, financial loss, time loss, denial of service,[2] and public discredit).

[1] We prefer this term over "identity theft" (IDT), although both have often been used [6,9,17]. 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.

S. Jajodia, and J. Lopez (Eds.): ESORICS 2008, LNCS 5283, pp. 130–145, 2008.

In the academic literature, there are relatively few proposals addressing IDF. Most focus on prevention of credential information extraction. (See, e.g. [3,4,14,26], for countermeasures to phishing or key logging.) We are aware of only one non-application-specific academic proposal addressing generic IDF [31], 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 involving real (vs. fictitious) people's identities. 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 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. While we deal with architectural problems associated with the design of privacy-preserving credential management systems, 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 [19] 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.

CROO provides means to both prevent IDF (by detecting IDF attempts), and limit its consequences when sophisticated IDF attacks have bypassed the aforementioned preventive measures. 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 few 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 [8,22], 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 to facilitate analysis and comparison of CROO with other proposals. §4 discusses related work. §5 concludes the paper.

2 Infrastructure and Protocol for IDF Detection

This section describes CROO, an infrastructure and protocol for IDF detection.

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-[3] 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.

[3] NFC [7] enables wireless communication between devices in very close (e.g. less than 10cm) proximity.

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.[4] 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.
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 channels.
5. Upon receiving $(ID_U, h(z), v)$, F uses ID_U to retrieve information required to compute $k^{(i)}$,[5] 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)

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

[5] 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 $k^{(i)}$.

Table 1. Notational Overview

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 2. Transaction Fraud Verification Protocol

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

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 $S_z = 2$;[6] (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,[7] 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),[8] 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

Driver's License. A real-world instantiation of CROO could be as follows: 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; D_U is a cell phone communicating with on-site terminals via NFC. (When validation of driver's licence information (e.g. for credit card issuing) does not currently involve online check with a trusted

[6] 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.

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

[8] 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).

party, this instantiation of our (online) proposal may be used to better detect driver's license-related IDF.)

Credit Card. As a second example, CROO can be instantiated with the following parties: 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; transaction details (e.g. dollar amount and R's identifier) are manually input by U into D_U; D_U displays v, and U manually inputs v (formatted as a credit card number with a 3-digit verification code) into a local PC used for web transactions.

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 Evaluation Criteria for Universal ID Fraud Solutions

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 below), rather than algebraically "proving" the security of S. Security- and privacy-related requirements of CROO are discussed in an extended version of this paper [24], as well as a preliminary mathematical security analysis of a simplified version of CROO. Devising realistic mathematical models and formal proofs which provide tangible guarantees in real-world deployments remains a challenge for us and others [16,15].

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-preserving capability (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 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. Due to space limitations, details of the evaluation claims are presented in an extended version of this paper [24].

Usability Evaluation Criteria

- ✓ **U1.** *No Requirement to Memorize Multiple Passwords.* S does not require U to memorize cred-token-specific 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).[9]
- ✓✗ **U3.** *No Requirement for Users to Carry Extra Devices.* S does not require U to carry extra personal devices (e.g. cell phone).
- ✓✗ **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).
- ✓ **U5.** *Support for Online Transactions.* S detects instances of attempted and/or committed but previously undetected IDF for online (e.g. web) transactions.
- ✓ **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.
- ✓ **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.
- ✓ **U8.** *Suitability for Fixed Users.* S can be used by fixed users (i.e. who carry out transactions from a constant geographic location).

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

✓ **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.

✗ **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).

✗ **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.

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.

✓✗ **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).

✓ **P3.** *No Disclosure of User Capabilities.* S does not reveal what hardware or software capabilities (e.g. digital camera or printer) U has.

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

Fraud Detection Evaluation Criteria

✓ **D1.** *Determination of Credential Use.* U and I know† when $x_U{}^*$ is used.

✓ **D2.** *Control on Credential Use.* U and I can control† the use of $x_U{}^*$ (i.e. approve or reject each use thereof).

✓ **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.

✓✗ **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.[10]

✓✗ **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).

✓ **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.[11]

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

[11] 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).

✓ **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).
✓ **D8.** *Verification of Credential Fraud Flag.* R can know whether x_U* is currently flagged as fraudulent.
✓ **D9.** *Detection of Clone Usage.* R (resp. I) can distinguish† x_U from its clones whenever the latter are presented to R (resp. I).
✗ **D10.** *Detection of Credential Cloning.* U and I can detect† that x_U* is cloned.
✗ **D11.** *Detection of Credential Theft.* U and I can detect† that x_U is stolen from U.
✓ **D12.** *Determination of Malicious Fraud Claims.* I can determine† whether U is honest when claiming that x_U* has been used without proper authorization.

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.
✓ **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).
✓ **C3.** *Protection Against Replay Attacks.* S prevents (or reduces to a negligible proportion) reuse of electronic messages sent to impersonate U.
✓ **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.
✗ **C5.** *Protection Against Denial of Service Attacks.* S prevents (or reduces to a negligible proportion) denial of services against U.

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[12] 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.

[12] 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.

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. A more extensive literature review is presented in an extended version of this paper [24].

Password-based Authentication. Static password schemes,[13] one-time password (OTP) schemes [18], password schemes resilient to shoulder surfing attacks [12,27], and schemes generating domain-specific passwords from a combination of single user-chosen passwords and multiple domain-specific keys [26,11] can all be used to authenticate users and thereby solve parts of the problem of phishing and/or IDF. Our scheme can be viewed as a careful combination of known and modified tools and techniques (e.g. cell phones, non-verifiable text [19], OTP-based authentication, and symmetric and asymmetric cryptography) to detect IDF.

Limited-Use Credit Card Numbers. Rubin and Wright [28] 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). Singh and dos Santos [30] 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 [29] 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 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. [23] propose a scheme for off-line generation of limited-use credit card numbers; their scheme is susceptible to dictionary attacks on user passwords.

Limiting the Effect of Cryptographic Key Exposure. Public-key schemes [5] have been proposed to limit the effect of key exposure by decreasing the odds that unauthorized public-key signatures be issued. Just and van Oorschot [13] suggest a method to detect fraudulent cryptographic signatures. CROO addresses the more general problem of IDF committed with cloned cred-info.

Device Capture Resilience. The idea of capture resilience was suggested by Mackenzie et al. [20] 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. In addition to differences in the way user passwords are generated in CROO and these schemes, CROO generates, for easier deployability, user credentials which can be

[13] Including commonplace typed textual password mechanisms, and strengthened password schemes [1].

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 and Personal Devices. Van Oorschot and Stubblebine [31] propose an IDF detection scheme for on-site transactions, whereby users' identity claims are corroborated with trusted claims of these users' location. Mannan and van Oorschot [21] propose an authentication protocol involving an independent personal device, and survey related schemes.

SET and Certificate-Based PKIs. SET [10] 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.

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 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. Abadi, M., Lomas, T.M.A., Needham, R.: Strengthening Passwords. Technical Report 1997 - 033, Digital Equipment Corporation (1997)
2. Balacheff, B., Chen, L., Pearson, S., Plaquin, D., Proudler, G.: Trusted Computing Platforms – TCPA Technology in Context. Prentice Hall, Englewood Cliffs (2003)
3. Chou, N., Ledesma, R., Teraguchi, Y., Boneh, D., Mitchell, J.C.: Client-Side Defense Against Web-Based Identity Theft. In: Network and Distributed System Security Symposium (NDSS 2004). The Internet Society (2004)
4. Dhamija, R., Tygar, J.D.: The Battle Against Phishing: Dynamic Security Skins. In: Symposium on Usable Privacy and Security (SOUPS 2005), pp. 77–88. ACM Press, New York (2005)
5. Dodis, Y., Franklin, M., Katz, J., Miyaji, A., Yung, M.: Key-Insulated Public Key Cryptosystems. In: Joye, M. (ed.) CT-RSA 2003. LNCS, vol. 2612, pp. 19–32. Springer, Heidelberg (2003)
6. 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)
7. NFC Forum (accessed, January 2008), http://www.nfc-forum.org/home
8. Goodrich, M.T., Sirivianos, M., Solis, J., Tsudik, G., Uzun, E.: Loud and Clear: Human-Verifiable Authentication Based on Audio. In: IEEE International Conference on Distributed Computing Systems (ICDCS 2006). IEEE, Los Alamitos (2006)
9. Gordon, G.R., Willox, N.A.: Identity Fraud: A Critical National and Global Threat. Journal of Economic Crime Management 2(1), 1–47 (2005)

10. Network Working Group. RFC 3538 - Secure Electronic Transaction (SET) Supplement for the v1.0 Internet Open Trading Protocol (IOTP) (2003) (accessed, January 2008), http://www.faqs.org/rfcs/rfc3538.html
11. Halderman, J.A., Waters, B., Felten, E.W.: A Convenient Method for Securely Managing Passwords. In: International Conference on World Wide Web (WWW 2005), pp. 471–479. ACM Press, New York (2005)
12. Haskett, J.A.: Pass-algorithms: A User Validation Scheme Based on Knowledge of Secret Algorithm. Communications of the ACM 27(8), 777–781 (1984)
13. Just, M., van Oorschot, P.C.: Addressing the Problem of Undetected Signature Key Compromise. In: Network and Distributed System Security (NDSS 1999). The Internet Society (1999)
14. Kirda, E., Kruegel, C.: Protecting Users Against Phishing Attacks with AntiPhish. In: Computer Software and Applications Conference 2005, pp. 517–524 (2005)
15. Koblitz, N., Menezes, A.J.: Another look at provable security II. In: Barua, R., Lange, T. (eds.) INDOCRYPT 2006. LNCS, vol. 4329, pp. 148–175. Springer, Heidelberg (2006)
16. Koblitz, N., Menezes, A.J.: Another look at provable security. Journal of Cryptology 20(1), 3–37 (2007)
17. Lacey, D., Cuganesan, S.: The Role of Organizations in Identity Theft Response: the Organization-Individual Dynamic. Journal of Consumer Affairs 38(2), 244–261 (2004)
18. Lamport, L.: Password Authentication with Insecure Communication. Communications of the ACM 24, 770–772 (1981)
19. Lomas, T.M.A., Gong, L., Saltzer, J.H., Needham, R.M.: Reducing Risks from Poorly Chosen Keys. ACM SIGOPS Operating Systems Review 23(5) (1989)
20. MacKenzie, P., Reiter, M.K.: Delegation of cryptographic servers for capture-resilient devices. Distributed Computing 16(4), 307–327 (2003)
21. Mannan, M., van Oorschot, P.C.: Using a personal device to strengthen password authentication from an untrusted computer. In: Dietrich, S., Dhamija, R. (eds.) FC 2007. LNCS, vol. 4886, pp. 88–103. Springer, Heidelberg (2007)
22. McCune, J.M., Perrig, A., Reiter, M.K.: Seeing-is-believing: Using camera phones for human-verifiable authentication. In: IEEE Symposium on Security and Privacy (May 2005)
23. Molloy, I., Li, J., Li, N.: Dynamic virtual credit card numbers. In: Dietrich, S., Dhamija, R. (eds.) FC 2007. LNCS, vol. 4886, pp. 208–223. Springer, Heidelberg (2007)
24. Nali, D., van Oorschot, P.C.: CROO: A Generic Architecture and Protocol to Detect Identity Fraud (Extended Version). Technical Report, TR-08-17, School of Computer Science, Carleton University, Ottawa, Canada (2008)
25. Javelin Strategy & Research. 2005 Identity Fraud Survey Report (2005), http://www.javelinstrategy.com/reports/2005IdentityFraudSurveyReport.html
26. Ross, B., Jackson, C., Miyake, N., Boneh, D., Mitchell, J.C.: Stronger Password Authentication Using Browser Extensions. In: USENIX Security Symposium, pp. 17–32 (2005)
27. Roth, V., Richter, K., Freidinger, R.: A PIN-Entry Method Resilient Against Shoulder Surfing. In: ACM Conference on Computer and Communications Security (CCS 2004), pp. 236–245. ACM Press, New York (2004)
28. Rubin, A.D., Wright, R.N.: Off-line generation of limited-use credit card numbers. In: Syverson, P.F. (ed.) FC 2001. LNCS, vol. 2339, pp. 196–209. Springer, Heidelberg (2002)

29. Shamir, A.: Secureclick: A web payment system with disposable credit card numbers. In: Syverson, P.F. (ed.) FC 2001. LNCS, vol. 2339, pp. 232–242. Springer, Heidelberg (2002)
30. Singh, A., dos Santos, A.L.M.: Grammar based off line generation of disposable credit card numbers. In: ACM Symposium on Applied Computing 2002 (SAC 2002), pp. 221–228. ACM Press, New York (2003)
31. van Oorschot, P.C., Stubblebine, S.: Countering Identity Theft through Digital Uniqueness, Location Cross-Checking, and Funneling. In: Patrick, A.S., Yung, M. (eds.) FC 2005. LNCS, vol. 3570, pp. 31–43. Springer, Heidelberg (2005)