# The Blinding of Weak Signatures (extended abstract)

Matthew Franklin*[1] and Moti Yung[2]

[1] C.W.I, Kruislaan 413, 1098 SJ Amsterdam
[2] IBM Research Division, T. J. Watson Center, Yorktown Heights, NY 10598

**Abstract.** The linearity of "check vectors" – a technique of secure distributed computation – gives an efficient solution to the problem of *blind weak signatures* (where a weak signature requires the on-line participation of a third party [17]). We refine aspects of the notion of "blinding a signature," and apply our weak schemes to on-line digital cash and other problems. The protocols we present are distinctly short, simple, and of low complexity.

## 1 Introduction

Blind signature schemes, as introduced by Chaum [4], allow a message holder to obtain a signature without disclosing the contents of the message to the signer. In this paper, we explore the possibility of blind signature without any cryptographic assumptions at all. This may seem an unlikely prospect, since any secure signature scheme – blind or otherwise – requires some intractability assumptions (one-way functions) [13]. However, what is true for standard signature schemes is not true for "weak" signature schemes.

Weak signatures were introduced by T. Rabin and Ben-Or [16] [17] to solve a problem (Verifiable Secret Sharing [8]) motivated by a question of general multi-party secure computation in the unconditional setting (network of untappable channels). They provide a form of authentication for which the on-line participation of a third party is needed. Check vectors are related to work on authentication codes [10] [19] and on universal classes of hash functions [3]. Using the idea of "check vectors," T. Rabin [16] showed that weak signatures can be implemented simply.

We show that weak signatures can be blinded easily (thus increasing their applicability). Actually, we identify several forms of blinding that are possible:

- **signature with blind verification** prevents the signer from later recognizing the signature, without necessarily hiding the message from him.
- **signature with blind message** prevents the signer from later recognizing the message being signed, without necessarily hiding the signature from him.
- **fully blind signature** combines blind verification and blind message in a single signature scheme (i.e., blinding in Chaum's original sense).

* This work was partially supported by an AT&T Bell Laboratories Scholarship

In this paper, we describe blinding schemes (of all three types) for weak signatures in the unconditional setting, and show how they can be used for digital cash and other applications with an on-line trusted center. Our methods exploit the linearity of check vectors in several ways. All of our signature blinding techniques are based on this linearity. Our on-line digital cash system takes advantage of linearity in a second way as well, to enable currency to be split easily into smaller denominations. An extension of our methods to multiple checking centers (to increase fault tolerance) can make use of linearity in yet a third way, since secure distributed computation of linear functions is cheap.

### 1.1 The On-Line Checking Center Approach

Our weak blind signature schemes, like the weak signature scheme on which they are based, rely on the presence of an on-line trusted server ("checking center"), separate from the signing agency and the message holder. This server participates in the creation of every signature, and also participates whenever a signed message holder wishes to prove to anyone that a signature is valid. The only functions performed by this trusted server are to store and retrieve information received from the signing agency and the message holder, and to compute certain linear combinations of values it receives. It can thus be a simple trusted device administered by a trusted authority. An on-line trusted server is the approach used in many practical distributed network security systems (e.g., Kerberos and KryptoKnight [14, 12]).

If the checking center is not trusted by the signing agency, then this assumption is not reasonable. In the schemes we describe, the checking center by itself can forge the signature of any message. Furthermore, the checking center can invalidate any signature by refusing to cooperate with the signed message holder during a validation request. However, as we will show, these vulnerabilities can be minimized by using multiple checking centers (as we explain in Section 4.3).

For blind signatures, the weakness of an on-line checking center is actually a strength in at least one sense: the possibility of traceability is maintained. We assume for security purposes that collaboration between the signer and the checking center will never occur. However, information held in their separate databases could in principle be combined to recover the message that was signed, e.g., as part of an authorized criminal investigation. This blend of security for the user and society is similar to that achieved by fair cryptosystems [15], where additional involvement of agents was suggested as a protection mechanism.

In another sense, the weakness of an on-line checking center is no weakness at all. One of the principle applications of blind signature schemes is to digital cash. Most digital cash schemes in the literature are "on-line," i.e., require that the bank be contacted for every transaction. Off-line digital cash schemes (introduced by Chaum, Fiat, and Naor [6]) are possible, but, because they can only detect certain abuses long after perpetration, they are often inapplicable (e.g., if an embezzler can reach a safe haven before being identified by the bank). Instead of contacting the bank for each transaction, on-line cash schemes that use our blind signatures contact the checking center for each transaction. The "cost" of

consulting the checking center for signature verification replaces the comparable cost of consulting the bank in a typical on-line cash scheme. In practice, many banks may be overseen by one central bank ("Federal Reserve"); the checking center could be managed naturally by the central bank to ensure overall control and integrity of the money supply (e.g., to prevent individual banks from exceeding their quotas for issuing new banknotes).

We conclude that the notion of a (somewhat) trusted checking agent – although inappropriate in some settings – is reasonable in various scenarios and systems; we assume such scenarios in the rest of the paper.

### 1.2 Organization of the Paper

In Section Two, we present some background and our model. Blind weak signatures based on check vectors are discussed in Section Three, along with applications. Section Four describes the use of blind weak signatures for digital cash.

## 2 Background Notions

### 2.1 Secure Computation Background

Secure distributed computation protocols enable a publicly known circuit to be jointly computed by a collection of processors, where each processor privately knows some of the inputs, such that certain properties of privacy and correctness are guaranteed despite some forms of misbehavior by the processors. Misbehavior may be passive (e.g., gossiping – but otherwise behaving as intended) or active (e.g., disrupting during the protocol in a coordinated manner).

After general cryptographic solutions appeared for the two-party case (by Yao [20]) and multi-party case (by Goldreich, Micali, and Wigderson [11]), more recent protocols have focused on the "unconditional" ("non-cryptographic") setting. In this setting, intractability assumptions are replaced by assumptions about the underlying communication model, e.g., that an untappable authenticated communication channel connects every pair of processors. General solutions in this model were given by Ben-Or, Goldwasser, and Wigderson [2] and by Chaum, Crépeau, and Damgård [5].

T. Rabin and Ben-Or [17] showed how increased protection against an active attack could be obtained in a (necessary) somewhat stronger communication model, i.e., adding a broadcast channel to an untappable network (see also [1]). These solutions depended on a strong secret sharing scheme due to T. Rabin [16], for which check vectors were originally designed.

It may be interesting to note that in this work we employ very efficiently tools from this area of "general secure distributed computation." Typically, this setting involves large communication overhead and has thus been doubted by practitioners. For example, Lampson, on this issue, has said: "...it has always been a complete mystery to me why anyone would ever want to do such a thing [general secure computation]." [9].

## 2.2 Check Vectors Background

In this subsection, we review check vectors [16] [17], which provide a type of "distributed error detection" combined with secrecy (note the relation to authentication codes). Specifically, the problem solved by check vectors is the following. There are three parties: a dealer $DLR$, a receiver $RCV$, and an intermediary $INT$. $DLR$ holds a secret $s \in Z_p$, and wishes to give $s$ to $INT$ so that $INT$ may later give $s$ to $RCV$. $RCV$ is said to "accept" $s$ from $INT$ if $RCV$ is convinced that this is the value $DLR$ originally sent to $INT$. Two properties must hold: (1) If $DLR$ and $RCV$ are honest, and if $DLR$ originally sends $s$ to $INT$, then $RCV$ will always accept $s$, and $RCV$ will reject $s' \neq s$ with high probability; (2) After $INT$ receives a value from $DLR$, $INT$ will know with high probability whether $RCV$ will subsequently accept that value.

A protocol that solves these properties is called an Information Checking Protocol. $RCV$ (or anyone else) would certainly accept the value $s$ from $INT$ (or from anyone else) if it were signed by $DLR$. Rabin gives the following solution. It assumes that untappable authenticated communication channels connect all pairs of participants.

*$DLR$ gives $s$ to $INT$*

1. $DLR$ chooses $a_1, b_1, y_1, \cdots, a_{2k}, b_{2k}, y_{2k} \in Z_p$ such that $a_i s + b_i = y_i \bmod p$ for all $i$, $1 \leq i \leq 2k$.
   (a) $DLR \rightarrow INT$: $s, y_1, \cdots, y_{2k}$.
   (b) $DLR \rightarrow RCV$: $a_1, b_1, \cdots, a_{2k}, b_{2k}$.
2. $INT \rightarrow RCV$: $i_1, \cdots, i_k \in \{1, \cdots, 2k\}$.
3. $RCV \rightarrow INT$: $a_{i_1}, b_{i_1}, \cdots, a_{i_k}, b_{i_k}$.
   (a) $INT$ verifies that $a_{i_j} s + b_{i_j} = y_{i_j} \bmod p$ for all $j$, $1 \leq j \leq k$.

*$INT$ gives $s$ to $RCV$*

1. $INT \rightarrow RCV$: $s, y_1, \cdots, y_{2k}$
   (a) $RCV$ accepts if $y_i = a_i s + b_i \bmod p$ for all $i$, $1 \leq i \leq 2k$.

It can be shown that the necessary properties of an Information Checking protocol hold in this case (where the high probabilities depend on $p$ and the security parameter $k$).

T. Pedersen has suggested that the same properties of check vectors can be achieved at a decrease of a factor of $k$ in communication complexity. Only two check equations are created by DLR, with $s, y_1, y_2$ going to INT and $a_1, b_1, a_2, b_2$ going to RCV. INT challenges RCV by sending a uniformly random $c \in Z_p^*$. RCV responds by sending $a' = ca_1 + a_2 \bmod p$ and $b' = cb_1 + b_2 \bmod p$. INT verifies that $a's + b' = cy_1 + y_2 \bmod p$. The probability of cheating remains small, i.e., $\frac{1}{p-1}$. The reduction in communication complexity extends to the Verifiable Secret Sharing protocol of Rabin [16] and the secure distributed computation protocols of Rabin and Ben-Or [17], as well as to the weak signature protocols in this paper.

**Notation**: We will often write $a, b, y$ to denote the corresponding vectors of check information.

### 2.3 Security Model

Our signature protocols have three parties. These parties usually will be referred to as a "message holder," a "signing agency," and a "checking center." We assume that messages can be sent between any pair of parties without any information about its contents being learned by any other participants; (in fact, no private messages are ever sent by the checking center).

The description of our signature protocols include a security parameter $k$ and a field size $p$. A weak signature protocol is secure if (1) the message holder cannot forge a signature without the collaboration of the checking center, except with very small probability (inversely proportional to $p$, and inverse exponentially proportional to $k$); and (2) cheating by the checking center or the signing agency which invalidates the signature will be detected by the message holder with high probability.

## 3 Blind Weak Signature

Here, we explain how check vectors enable a form of weak signature, and we describe how to modify the basic Information Checking Protocol so that the weak signatures are blinded. We assume throughout this section that all pairs of participants are connected by an untappable authenticated communication channel.

### 3.1 Check Vectors Give Weak Signature

Rabin's Information Checking Protocol gives a weak signature scheme. Consider that the intermediary $INT$ wishes to have a message $s$ signed by the dealer $DLR$. If $INT$ gives the message $s$ to $DLR$, then the first phase of the Information Checking Protocol has the following effect. The original message holder $INT$ ends up with the "signed message" $s, y$, while a third party $RCV$ ends up with the check information $a, b$. Anyone can determine the validity of the signature by asking $RCV$ to reveal the check information; this request, and its reply, do not need to be sent through private channels. The signature is weak, because the assistance of this third party is needed to verify a signature. More generally, any authentication code [10] [19] can be used as a weak signature scheme by giving the message and tag to the original message holder and the key to the third party.

Notice that the Information Checking Protocol can be modified so that the Receiver ends with many check vectors $a, b$ for the same signed message $s, y$. Each subsequent request for signature validation can be met by revealing a new set of check information. Forgery by a message holder would be possible if check information were reused.

Notice also that the Receiver never needs to see the messages it is validating. Suppose the signing Dealer sends the same random tag to both the message holder $INT$ and the validator $RCV$ in Step 1 of the Information Checking Protocol. Each request for signature validation can then be indexed by this random tag, so $RCV$ knows which check pairs to reveal.

We remark that another signature scheme in the unconditional setting was introduced by Chaum and Roijakkers [7]. It satisfies a stronger set of conditions than Rabin's Information Checking Protocol, at a great increase in communication cost.

### 3.2 Weak Signature with Blinded Verification

One aspect of blind signature schemes is that the signer should not be able to connect a signature to the protocol that produced it without collaboration from another participant in the signing protocol. We call a scheme with this property a signature scheme with "blinded verification."

**Claim 1.** *There exists a weak signature scheme with blinded verification.*

Due to the linearity of the basic check equation $y = as + b \bmod p$, it is easy for $INT$ and $RCV$ to blind the signature $y$ and check information $a, b$ so that they will be unrecognizable to $DLR$. $INT$ sends to $RCV$ uniformly random vectors of offsets $\Delta a, \Delta b \in_R Z_p^j$, where $j$ is the length of $a$ and $b$. $RCV$ finds $a' = a + \Delta a \bmod p$, and $b' = b + \Delta b \bmod p$. $INT$ finds $y' = y + s\Delta a + \Delta b \bmod p$. The check equation remains valid: $y' = y + s\Delta a + \Delta b = (as + b) + s\Delta a + \Delta b = (a + \Delta a)s + (b + \Delta b) = a's + b' \bmod p$.

This means that weak signature is possible such that the signature is later unrecognizable to the signer. Although the message $s$ is unchanged, the vectors $y, a, b$ are replaced by uniformly random vectors that satisfy that check equation. The signer will be unable to distinguish two identical messages that were signed at different times.

### 3.3 Weak Signature with Blinded Message

Another aspect of blind signature schemes is that the message itself should be concealed from the signer in the absence of collaboration from another party in the signing protocol. We call a scheme with this property a signature scheme with "blinded message."

**Claim 2.** *There exists a weak signature scheme with blinded message.*

The linearity of the basic check equation $y = as + b \bmod p$ also makes this form of blinding easy to achieve. Suppose that $INT$ wishes to have a message $s$ signed by the dealer $DLR$. $INT$ randomly chooses $r \in_R Z_p^*$, and sends $rs \bmod p$ to $DLR$. The parties now run the first phase of the Information Checking Protocol on $rs \bmod p$. The checking center $RCV$ ends up with $a, b$, and the

message holder $INT$ ends up with $y$, such that $y = ars + b \bmod p$. Now $INT$ sends $r$ to $RCV$, and $RCV$ computes $a' = ar \bmod p$ (or invalidates the signature if $r$ is not sent promptly). $RCV$ stores the check information $a', b$, and the check equation holds: $y = a's + b \bmod p$.

### 3.4 Fully Blinded Weak Signature

We call a signature scheme "fully blind" if it is both verification blinding and message blinding; this is blinding in Chaum's original sense. Combining the techniques of the preceding two subsections yields a fully blind weak signature scheme. At the point in the message blinding scheme where $INT$ would send $r$ to $RCV$, $INT$ sends instead $r, \Delta a, \Delta b$ to $RCV$. $INT$ computes $y' = y + s\Delta a + \Delta b \bmod p$. $RCV$ computes $a' = ar + \Delta a \bmod p$ and $b' = b + \Delta b \bmod p$. The check equation still holds: $y' = a's + b' \bmod p$.

**Claim 3.** *There exists a fully blind weak signature scheme.*

### 3.5 Applications and Cost of Weak Blind Signature

Weak blind signatures can replace blind signatures in an application if it reasonable to include an on-line trusted checking center. One application is for pseudonymous credentials, allowing a user to establish different identities with different organizations. Another application is timestamping, allowing a user to associate a digital document with its time of creation; since such documents often become public when the timestamp is verified, signature with blind message may suffice. A third application area is for anonymous access control schemes and digital cash schemes, where signed messages are tokens that can be exchanged for some product or service; when only a few types of token are in circulation, signature with blind verification may suffice.

For all of these applications, the complexity of the protocol is distinctly low. The signing agency performs one simple linear computation, and stores nothing. The message holder performs one simple linear computation, and stores a $k$-tuple of elements (in $Z_p$) as large as the element being signed (e.g., the size of an access token or random tag, or the size of a hash of a document). The checking center performs one simple linear computation, and stores two $k$-tuples of checking elements for later retrieval. The signing protocol takes one and a half rounds, where no message is longer than two $k$-tuples of elements. The checking procedure is another simple linear computation. The signing agency and message holder each need to select uniformly random elements of $Z_p$ for the signing protocol, which can be done efficiently in practice using cryptographically strong pseudorandom generators.

## 4 Weak Digital Cash

A digital cash scheme [4] is a set of cryptographic protocols for withdrawal (by a customer from the bank), transfer (by a customer to a vendor or another

customer), deposit (by a vendor to the bank), and also division (by a customer), such that the security needs of all participants are satisfied: anonymity of use and assurance of authenticity for customers, and impossibility of undetected reuse or forgery for the bank.

A cash scheme is "on-line" if additional agents (e.g., bank or checking center) must participate in transfer or division protocols. Weak signature with blinded verification can be used to implement an on-line digital cash scheme, in which the checking center is consulted for each purchase. We call such a scheme a "weak digital cash scheme."

We consider schemes that require at least one on-line checking center in addition to the bank. No cryptographic assumptions are required, other than untappable channels between pairs of participants. Our scheme relies on weak signature with blinded verification, and does not require the power of fully blind weak signatures.

### 4.1 Why Sign At All?

Simple schemes – without any form of signature – are possible given a second trusted agency. For example, something like a numbered "Swiss bank account" can be set up at the checking center, which then authorizes every purchase directly. The bank issues an identifying random tag (pseudonym) to the customer when money is withdrawn, and an account using that tag is set up at the second agency. The customer can then "refresh" the random tag at the second agency (by sending the old random tag together with a new one), and then have all transactions be essentially simple withdrawals and deposits from this "numbered" account.

However, the second agency learns the size of all purchases and transfers in this type of scheme. This presents temptations that could be problematic, and also threatens anonymity of money users. In an automatic toll collection scheme, for example, purchases correspond to distance traveled, which may be sensitive information. We would like a scheme where the on-line agency learns nothing about the size of any transactions.

### 4.2 Weak Signature with Blinded Verification Hides Amounts

Using weak signature with blinded verification, it is possible to have a digital cash scheme in which the checking center learns nothing about money amounts of any transactions.

**Claim 4.** *There exists a weak digital cash scheme supporting anonymity, security, unlimited transferability, and divisibility.*

To withdraw a $d$ unit coin, a customer ($INT$) gets a signature of $d$ from the bank ($DLR$) with the help of an on-line checking center($RCV$), where the amount of check information given to $RCV$ determines the number of transactions possible with this coin. The customer and the on-line checking center then

blind the signature and check information with offsets, as described in the section on weak signature with blinded verification. To transfer a coin (e.g., make a purchase), the old coin holder convinces the new coin holder that the signature of $d$ is valid by revealing some of the check information; the new coin holder then takes over the money by reblinding signature and remaining check information. To make a deposit, the current coin holder gives it to the bank, and asks the on-line check center to send the remaining check information to the bank as well; the bank accepts the deposit if the check equations hold. Throughout the lifetime of a coin, the checking center indexes its check information by a random tag which is also known to the coin holder, initially assigned by the bank, and updatable by the new coin holder for transfers.

Divisibility of coins is simple as well, again due to the linearity of the basic check equation. The holder of a coin of value $d$ splits each check signature $y$ into two check signatures $y/2 \bmod p, y/2 \bmod p$. The coin holder also asks the check center to split each check pair $[a, b]$ into two check pairs $[a, b/2 \bmod p]$, $[a, b/2 \bmod p]$. All check equations still hold for the two split coins of value $\frac{d}{2}$. Now signature and check information for both split coins are reblinded.

However, care must be taken to prevent a one unit coin from being split into two $\frac{p+1}{2}$ unit coins. One way to prevent this type of cheating is to assume that the modulus of the check equations is a large prime $p = 2q + 1$, where $q$ is itself prime, and that all withdrawals are for $d = 2^k$ units, where $0 \leq k < \log_2 p$. This restriction of allowable denominations implies that a cheating coin holder would need to perform an infeasible number ($O(p - \log p)$) of splits to produce a valid larger denomination from a one unit coin.

### 4.3 Multiple Checking Centers

In the money scheme described above, each consumer runs the risk that the checking center will forget the check pairs or tag values for some of its money. The bank runs the risk that the checking center will create its own money through forgery. Protection against these abuses is straightforward if there is more than one checking center. Each checking center can be given its own check pairs, and they all can participate in money transactions. If majority agreement is necessary for every transaction, then no minority of checking centers can cause a consumer to lose money that rightfully belongs to it, and no minority can cheat the bank through forgery. There is no need for the checking centers to communicate among themselves.

**Claim 5.** *There exists a weak digital cash scheme with multiple checking centers which is secure against active cheating by any minority of checking centers.*

To protect the user's privacy against collaboration between bank and checking center, multiple checking centers could use secure distributed computation protocols [2] [5] [17] to simulate a single check center. This would require untappable channels among the checking centers, but little communication since all secure computations are for *linear operations*. Thus employing a small number

of distributed checking centers (e.g., three or five) to compute together the role of a centralized one still yields a very practical system.

**Acknowledgements:** We thank Torben Pedersen and members of the Eurocrypt Program Committee for their helpful comments.

## References

1. D. Beaver, "Distributed computations tolerating a faulty minority, and multiparty zero-knowledge proof systems," J. Cryptology 4, 2 (1991).
2. M. Ben-Or, S. Goldwasser, and A. Wigderson, "Completeness theorems for non-cryptographic fault-tolerant secure distributed computation," ACM STOC 1988.
3. J. Carter and M. Wegman, "Universal classes of hash function," JCSS 18 (1979), 143–154.
4. D. Chaum, "Security without identification: transaction systems to make big brother obsolete," CACM 28, 10 (October 1985).
5. D. Chaum, C. Crépeau, and I. Damgård, "Multiparty unconditionally secure protocols," ACM STOC 1988.
6. D. Chaum, A. Fiat, and M. Naor, "Untraceable electronic cash," Crypto 1988, 319–327.
7. D. Chaum and S. Roijakkers, "Unconditionally secure digital signatures," Crypto 1990, 206–214.
8. B. Chor, S. Goldwasser, S. Micali and B. Awerbuch, "Verifiable secret sharing" IEEE FOCS 1985, 383–395.
9. J. Feigenbaum and M. Merritt, "Open questions, talk abstracts, and summary of discussions," in *Distributed Computing and Cryptography*, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, Vol. 2, 1991, 1–45.
10. E. Gilbert, F. MacWilliams, and N. Sloane, "Codes which detect deception," Bell Systems Technical Journal 53 (1974), 405–424.
11. O. Goldreich, S. Micali, and A. Wigderson, "How to play any mental game," ACM STOC 1987, 218–229.
12. A. Herzberg, P. Janson, S. Kutten, R. Molva, G. Tsudik and M. Yung, "KryptoKnight: light-weight authentication and key distribution protocols," Manuscript.
13. R. Impagliazzo and M. Luby, "One-way functions are essential for complexity based cryptography," IEEE FOCS 1989, 230–235.
14. J. Kohl, "The use of encryption in Kerberos for network authentication," Crypto 1989, 35–43.
15. S. Micali, "Fair public-key cryptosystems," Crypto 1992.
16. T. Rabin, "Robust sharing of secrets when the dealer is honest or cheating," M.Sc. Thesis, Hebrew University, 1988.
17. T. Rabin and M. Ben-Or, "Verifiable secret sharing and multiparty protocols with honest majority," ACM STOC 1989, 73–85.
18. R. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," CACM 21 (1978), 120–126.
19. G. Simmons, "Authentication theory / Coding theory," Crypto 1984, 411–432.
20. A. Yao, "How to generate and exchange secrets," IEEE FOCS 1986, 162–167.