JOURNAL OF COMPUTER AND SYSTEM SCIENCES 27, 256–267 (1983)

Transaction Protection by Beacons

MICHAEL O. RABIN*

Department of Mathematics,
The Hebrew University of Jerusalem, Jerusalem, Israel and
Department of Computer Science,
Harvard University, Cambridge, Massachusetts 02138

Protocols for implementing contract signing, confidential disclosures, and certified mail in an electronic mail system are proposed. These transactions are provably impossible without a trusted intermediary. However, they can be implemented with just a small probability of a participant cheating his partner, by use of a beacon emitting random integers. Applications include privacy protection of personal information in data banks, as well as the protection of business transactions.

1. INTRODUCTION

Digital signatures enable a person to prepare and sign a document for transmission through electronic mail [3, 4, 6]. Digital signatures, however, do not by themselves provide procedures for protection of transactions involving more than one person, which the business community of users of electronic mail needs to perform. Of particular interest are the signing of contracts, disclosure of confidential information, and certified mail.

Assume that Alice and Bob negotiate over the telephone a contract CONT stating that Alice will sell her house to Bob for a certain price. If Alice signs CONT (say by use of a digital signature) and sends it to Bob, then he has the option of delaying the return of CONT signed by him while he looks around for a better/cheaper house. During this time Alice is already committed to the deal.

The usual method for overcoming this problem, when Alice and Bob cannot actually get together, is to use a trusted intermediary who gets Alice's signed contract and holds it until he gets CONT signed by Bob, at which time he sends to Bob and Alice the appropriate signed copies.

A disclosure protocol is needed when Alice and Bob have agreed that Alice will disclose to Bob her secret process D for the manufacture of widgets, and Bob promises not to divulge D. This agreement mentions D but does not detail it. Such a confidential disclosure is often intended to enable Bob to examine D so that he can decide whether to buy it. When Alice later on reveals D to Bob she will need a

* This research was supported in part by NSF Grant MCS-80-12716 at the University of California at Berkeley and by NSF Grant MCS-8121431 at Harvard University.

0022-0000/83 $3.00

receipt from him, for otherwise she cannot enforce the agreement. However, Bob can stop communicating with Alice once he has received D, leaving her without a receipt.

Confidential disclosure protocols may become important for privacy protection procedures in which a person or an organization are given personal information such as medical records on condition that they keep the information confidential.

Mail certification allows Alice to send a message M ("A ton of widgets on its way to you") to Bob with the assurance that she will have a receipt. This differs from disclosures in that Bob may be allowed to examine M before deciding wether he wants to accept it.

The Post Office, acting as a trusted intermediary, provides certified return-receipt mail. Note, however, that Alice can only obtain a receipt stating that Bob has received a letter from her. The content of the letter is not specified.

The above transactions can be implemented in an electronic mail system by establishing message exchange centers which will act as trusted intermediaries for every transaction. There are many difficulties involved in this approach. First, the centers will be foci for a very heavy traffic load. The center will have to maintain an extensive data base to keep track of the transactions for which it acts as intermediary. Errors are unavoidable and this raises the question of liability of the intermediary.

We propose a solution employing a "beacon" which emits at regularly spaced time intervals, randomly chosen integers in the range $1 \leqslant i \leqslant k$. The beacon can be a channel in a communications satellite or a node in a network which can be accessed by every participant. The volume of messages emanating from the beacon is actually very low and the users have only to listen, so that there will be no bottlenecks.

It is easy to see that certified mail is impossible without some form of trusted intermediary, and similarly for the other transactions. Thus it is natural to resort to randomized algorithms which will provide a solution, while allowing a certain tolerable probability of cheating.

Our main result is that with a beacon emitting random integers $1 \leqslant i \leqslant k$ spaced apart at time intervals Δ, we can have protocols for the signing of contracts, disclosures, and certified mail, which require an expected exchange of $2k$ messages per transaction, and will end in expected time $k\Delta$. Furthermore, the probability of one participant in the transaction cheating another is at most $1/k$. The value $1/k$ is actually best possible for *any* randomized protection protocol for those transactions which involves an exchange of $2k$ messages.

Actually we shall use two somewhat different versions of the beacon, a *signature beacon* for signing contracts and for certified mail, and a *disclosure beacon* for implementing confidential disclosures.

The use of randomized protocols for transaction protection was first introduced in [5]. The articles [1, 2] contain different (randomized) solutions for some of the problems treated here, as well as a proof for the above-mentioned lower bound on the probability of cheating. The author wishes to thank Manuel Blum for many fruitful conversations on the topics of this paper.

2. Public Key Systems

We assume throughout this paper the existence of a public key digital signature and encryption system, see [3, 4, 6].

This involves functions E_x and D_y depending on parameters x and y such that for every encoding key K there exists a decoding key $d(K)$ so that for any message M we have $D_{d(K)}(E_K(M)) = M$.

The assumption is that knowledge of K does not provide knowledge of the decoding key $d(K)$. On the other hand, the encoding key K can be trivially computed from the decoding key. For example, with Rivest *et al.* [6], we adopt the convention that the encoding key is a pair $K = (e, n)$, where n is a product of two large primes and $E_K(M) \equiv M^e \bmod n$. The decoding key is (d, p, q), where $n = p \cdot q$, p, q primes and $ed \equiv 1 \bmod (p-1)(q-1)$. In [4] the encoding key is $K = (b, n)$ with n as above, and $E_K(M) \equiv M(M+b) \bmod n$. The decoding key is (b, p, q).

Each participant P publishes a key selected by him, call it also P, but keeps the decoding key $d(P)$ to himself. Actually, key selection proceeds by selecting $d(P)$ and constructing P.

If the mapping $M \rightarrow E_K(M)$ is essentially an onto mapping, then the public key system can also be used to digitally sign a message. We shall denote P's signature on M by $S_P(M)$. Sometimes we shall use "*M*. signed *P*" to indicate the message M digitally signed by P. For example "Will send a ton to Bob. signed Alice."

3. The Beacon

We shall require two versions of the beacon, one for contract signing and the other for confidential disclosures. The beacon will have a public key BN and will broadcast at time intervals Δ. It will time-stamp its messages with the full time of transmission. To prevent ambiguity, time is Greenwich Mean Time. Thus if $\Delta = 3$ seconds, then $t =$ 10.19.1981, 14:39:54 is a possible time stamp, and $t + \Delta =$ 10.19.1981, 14:39:57.

Let $i :=$ random mean that i is a randomly selected integer in the interval $[1, k]$. A *signature beacon* behaves according to the following protocol.

```
Signature Beacon's Protocol
select some future time $t_0$;
$t := t_0$;
while true do
  begin
    $i :=$ random; *do before time $t$*
    at time $t$ broadcast: $S_{BN}(i, t)$;
    $t := t + \Delta$
  end
```

In order to implement disclosures we need an operation $p :=$ randomkey which means that a *decoding* key $d(p)$ is randomly chosen and p is assigned the value of the corresponding public encoding key. Thus the process performing $p :=$ randomkey knows the public key p as well as the private decoding key $d(p)$.

```
Disclosure Beacon's Protocol
select a future time $t_0$;
$t := t_0$;
while true do
  begin
    $j :=$ random; *do before time $t$*
    $d := d(p_j)$; *do before time $t$*
    $p_1 :=$ randomkey,..., $p_k :=$ randomkey; *do before time $t$*
    at time $t$ broadcast: "$p_1$,..., $p_k$, $d$, $j$, $t$. signed BN";
    $t := t + \Delta$
  end
```

Note that at time t, the disclosure beacon broadcasts k public keys for which it knows the decoding keys, and the decoding key d for a key randomly selected out of those broadcast at time $t - \Delta$.

4. Contract Signing

Let us start by considering the situation that Alice and Bob have discussed a contract and want to sign it. Let CONTR denote the actual string of characters which is the text of the contract and let C be some name for the contract, selected by Alice and Bob. In practice C will be a number obtained by applying some hash function h to the string denoted by CONTR, i.e., $C = h(\text{CONTR})$.

The participants will use the signature beacon. Messages of the form $M = (i, t)$ will be called *beacon messages* and the broadcasts $S_{\text{BN}}(M)$ will be called *beacon signals*.

Alice starts by sending Bob a signed message:

P_A = "If Bob can produce (C, M) signed by Alice and M signed by the beacon, for some beacon message M, then I, Alice, will be commited to "CONTR" as of the time t mentioned in M. signed Alice." (1)

Note that Alice is signing the actual string preceding her signature, except that C and CONTR are replaced by the number and character string which they denote.

Bob sends Alice a similarly constructed message P_{B} with the names "Alice" and "Bob" interchanged, and signed by him. If Alice does not receive this message, then she does not continue with the protocol. In this case nothing has happened because Bob does not have Alice's signature on the contract.

Let us call the above two signed messages P_{A} and P_{B} the *preliminary agreements*.

Now for the actual protocol for the exchange of signatures. We assume that each participant, including the beacon, has a clock, and that all of these clocks are synchronized. Clock readings are always taken at starting points of the beacon's time intervals. The notation $t :=$ clock means that t is assigned the nearest future clock reading. Recall that A and B are, respectively, Alice's and Bob's public keys.

```
Alice's Protocol
(1)  willing := true;
(2)  send $P_A$ to Bob;
(3)  if received $P_B$ then $t :=$ clock; *synchronized with Bob*
(4)  if received $P_B$ then while willing = true do
     begin
(5)    $i_A = $ random;
(6)    send $i_A$ to Bob; *after time $t$, before $t + \Delta/6$*
(7)    if received $i_B$ from Bob then

(8)        $i = i_A + i_B \bmod k$;
(9)        send Bob $S_A(C, i, t + \Delta)$; *before $t + \Delta/3$*

(10)   if Bob has not returned $S_B(C, i, t + \Delta)$ before time $t + 2\Delta/3$ then
       willing := false;
       wait until time $t + \Delta$;
       if beacon signal is $S_{BN}(i, t + \Delta)$ then willing := false;
       $t := t + \Delta$
     end
```

Bob's protocol is exactly like Alice's with "A" and "Alice" replaced by "B" and "Bob" and vice versa.

Comment 1. Alice and Bob must start with assignments $t :=$ clock resulting with the same value for t. A special protocol is required to ensure this synchronization; we omit the details.

Comment 2. A statement such as "if Bob ...$S_B(C, i, t + \Delta)$..." in Alice's protocol means that she actually checks Bob's digital signature on the message received from him. The time intervals available are sufficient for these computations.

If Alice and Bob behave according to the above protocols, then within an expected number k of time-intervals, the integer i randomly chosen by Alice and Bob will be the one contained in the next beacon message at a time t_1. When this happens both Alice and Bob will have valid time-stamped signatures on the contract.

If a dispute arises and, say, Alice denies having signed the contract, then Bob can take her to court and prove that she has signed. Namely, Bob will produce the preliminary agreement P_A detailed in (1) and the signed messages $H = S_A(C, i, t_1)$, $G = S_{BN}(i, t_1)$. All the judge has to do is verify Alice's digital signatures in P_A and H, and the beacon's signature in G. Thus the adjudication of a dispute concerning signatures involves just ordinary digital signature verification. The judge need not concern himself with the protocols by which P_A, H, and G were created. These protocols are followed by Alice and Bob for their own protection and ensure the following result.

THEOREM 1. *If Alice and Bob follow their respective protocols, then after expected time $k\Delta$ each will have the other's binding signature on the contract.*

If Alice follows her protocol and Bob tries to cheat her, then his probability of obtaining her signature on the contract, without Alice having his signature, is $1/k$. Bob's probability of being caught, resulting in the contract not being signed at all, is $1-1/k$.

The same statement holds for Bob's probability of being cheated.

Proof. Assume that Alice and Bob start their protocols at time t. Within each time interval $[t+(n-1)\Delta,\ t+n\Delta]$ they exchange messages $S_A(C, i, t+n\Delta)$ and $S_B(C, i, t+n\Delta)$ for some randomly chosen $i=i_n$, $1 \leqslant i \leqslant k$. The probability that at time $t+n\Delta$ the beacon's signal will be $S_{BN}(i, t+n\Delta)$ is $1/k$. Thus the protocol will end within expected time $k\Delta$. As explained just before the theorem, when the protocol ends, each participant has the other's binding and time-stamped signature on the contract.

Assume that Alice follows her protocol and that Bob does not respond in some time interval $[t+(n-1)\Delta, t+n\Delta]$ with $S_B(i, t+n\Delta)$, where $i=i_n$ is the integer used by Alice in that interval. Bob will have Alice's signature only if the beacon's signal is $S_{BN}(i, t+n\Delta)$ for the same i, and this will occur with probability $1/k$.

If Bob does not respond properly then Alice stops, so that his probability of not obtaining a signature at all is $1-1/k$.

It is clear that Alice's and Bob's roles in the protocol are completely symmetrical. ∎

The above two-party protocol can be readily extended to n participants $A_1,..., A_n$ who wish to jointly sign a contract. Each participant A_i starts by signing a preliminary agreement

> P_{A_i} = "If anyone of $A_1,..., A_{i-1}, A_{i+1},..., A_n$ can produce (C, M) signed by A_i and M signed by the beacon, for some beacon message M, then I, A_i, will be comitted to CONTR as of the time t mentioned in M.
> signed A_i."

The protocol for A_i is obtained from Alice's protocol by changing A into A_i, i.e., Alice into A_i, S_A and i_A into S_{A_i} and i_{A_i}. Line 2 is changed to "Send P_{A_i} to every A_j, $j \neq i$..." Line 3 is changed to "if received every $P_{A_j}, j \neq i$...". Line 10 is changed to "if *not every* A_j, $j \neq i$, has returned $S_{A_j}(C, i, t+\Delta)$...". In line 7 have $i=\sum_1^n i_{A_j} \bmod k$; etc.

It is clear that with the appropriate changes, Theorem 1 holds for the case of n participants.

5. Confidential Disclosures

We now consider the situation that Alice has a secret DIS and has agreed with Bob to divulge it to him and he has agreed to keep it confidential. To enforce this agreement Alice must be sure that when she discloses the secret to Bob, she will have his receipt for DIS. The difference between disclosure in return for a receipt and contract signing is that in the latter case the parties know what the contract is and just have to insure that everybody signs. With the disclosure, Bob does not know what the disclosure DIS is, and DIS itself cannot be used in the protocol in clear, i.e., nonencrypted, form.

We shall use the *disclosure beacon* to implement a disclosure protocol.

Let A and B be Alice's and Bob's public keys, so that $d(\mathrm{A})$ and $d(\mathrm{B})$ are the corresponding private decryption keys.

Alice and Bob agree on a reference number *dis* for the disclosure DIS. Alice chooses an encryption key K. She doubly encodes DIS by Bob's key B, followed by K, to produce $M = E_K(E_B(\mathrm{DIS}))$. Alice now sends Bob a signed message

N_{A} = "M is the result of my disclosure, referred to by *dis*, being encoded by B followed with encoding by K. signed Alice." (2)

The idea of the disclosure protocol is that at time t Alice and Bob select an integer $1 \leqslant i \leqslant k$. Bob sends Alice the signed message $S_{\mathrm{B}}(dis, i, t+\varDelta)$. Alice then sends Bob the decoding key $d(K)$, encrypted by the ith key p_i released by the beacon at time t, i.e., $E_{p_i}(d(K))$. If at time $t+\varDelta$ the beacon releases the decoding key $\bar{d} = d(p_i)$ for the same key p_i, then provided that Alice behaved according to the protocol, Bob can decode: $D_{\bar{d}}(E_{p_i}(d(K)) = d(K)$. He can then compute $E_{\mathrm{B}}(\mathrm{DIS}) = D_{d(K)}(M)$ and $\mathrm{DIS} = D_{d(B)}(E_{\mathrm{B}}(\mathrm{DIS}))$.

Recall that the disclosure beacon's signal at time $t+\varDelta$ has the form

"$\bar{p}_1, \ldots, \bar{p}_k$, $\bar{d}$, $\bar{j}$, $t+\varDelta$ signed BN", where $\bar{d}$ is the decoding key for the p_j broadcast at time t. (3)

By prior agreement, if Bob has sent $S_{\mathrm{B}}(\mathrm{dis}, i, t+\varDelta)$ and the beacon's signal (3) at time $t+\varDelta$ satisfies $\bar{j} = i$, i.e., $\bar{d}$ is in fact the decryption key for the ith key p_i broadcast at time t, then Bob admits to having received DIS unless he can prove that the key sent to him by Alice is not $d(K)$.

Now for the detailed protocols. Alice and Bob start by signing an agreement stating that Alice will disclose to Bob information (the disclosure) having certain properties. Bob agrees that if he receives the disclosure described by Alice, then he will keep it confidential and not use it without Alice's consent. They agree to refer to the disclosure by the number *dis*. The above agreement will be called the ***disclosure agreement***. Note that the disclosure agreement does not detail the disclosure itself.

Alice now gives Bob her signed obligation N_{A}, see (2). Bob prepares the following signed message:

N_B = "If Alice can produce for some time t and for some i message of the form "*dis*, *i*, $t+\Delta$. signed Bob" and "$\bar{p}_1, ..., \bar{p}_n, \bar{d}, i, t+\Delta$. signed BN", then I am committed to having received the disclosure referred to by *dis*, unless I can produce a message of the form "*dis*, X, i, t. signed Alice" such that $D_{\bar{d}}(X)$ is not decoding key $d(K)$ for the key K. signed Bob." (4)

Bob sends N_B to Alice. To avoid confusion, let us point out that in the above, *dis* and K are specific numbers (K may be a sequence of numbers). However, $d(K)$ is not a number but rather a string such as $d(30110281)$.

```
Bob's Disclosure Protocol
  willing := true;
  if received $N_A$ from Alice then send her $N_B$;
  if received $N_A$ then $t$ := clock; *synchronized with Alice*
  if received $N_B$ then while willing = true do
    begin
      $i$ := random; *after time $t$ and before $t+\Delta/6$*
      send $i$ to Alice; *after time $t$ and before $t+\Delta/6$*
      send to Alice $S_B(dis, i, t+\Delta)$; *before time $t+\Delta/3$*
      if not received from Alice a message $S_A(\text{dis}, X, i, t)$ before
        time $t+2\Delta/3$ then willing := false;
      wait until time $t+\Delta$;
      if Beacon signal is $S_{BN}(\bar{p}_1, ..., \bar{p}_k, \bar{d}, i, t+\Delta)$ then willing := false;
      $t := t+\Delta$
    end
```

```
Alice's Disclosure Protocol
  willing := true;
  $i := 0$;
  send $N_A$ to Bob;
  if received $N_B$ then $t$ := clock; *synchronized with Bob*
  if received $N_B$ then while willing = true do
    begin
      if received $i'$ from Bob and beacon signal is $S_{BN}(p_1, ..., p_k, d, j, t)$
      then $i := i'$ and $X := E_{p_l}(d(K))$;
      if received $S_B(dis, i, t+\Delta)$ from Bob before time $t+\Delta/3$ then
      send to Bob $S_A(dis, X, i, t)$ before time $t+2\Delta/3$;
      if at time $t+\Delta$ beacon signal is $S_{BN}(\bar{p}_i, ..., \bar{p}_k, \bar{d}, i, t+\Delta)$ then
      willing := false;
      $t := t+\Delta$
    end
```

The disclosure protocol has the following properties:

THEOREM 2. *If Alice and Bob follow their protocols, then within expected time $k\Delta$, Bob will have the disclosure and Alice will have Bob's receipt for it.*

If Alice follows her protocol, then regardless of what Bob does, if he receives the disclosure, then Alice will have his receipt. Also, Bob cannot cause the disclosure to be revealed to a third party before Alice has his receipt.

If Bob follows his protocol and Alice tries to cheat him, then her probability of success is $1/k$. Her probability of being caught is $1 - 1/k$, in which case Bob will terminate the exchange.

Proof. Assume that Alice and Bob follow their respective protocols. Let t be a beacon time in the execution of the protocols. Bob selects an $1 \leqslant i \leqslant k$ and sends $G = S_{\mathrm{B}}(dis, i, t+\Delta)$ to Alice. She responds by sending $S_A(dis, X, i, t)$, where $X = E_{p_i}(d(K))$ and p_i is the ith encryption key sent by the beacon at time t. After a Δ interval, at time $t+\Delta$, the beacon signals

$$H = S_{\mathrm{BN}}(\bar{p}_1, \ldots, \bar{p}_k, \bar{d}, j, t+\Delta), \qquad (6)$$

where $j = $ random.

After an expected number k of repetitions, $i = j$ will occur. When this happens then $\bar{d} = d(p_i)$, and Bob can find $d(K) = D_{\bar{d}}(X)$ and $\mathrm{DIS} = D_{d(\mathrm{B})}(D_{d(K)}(M))$, see (2) for the meaning of M. Thus the expected total elapsed time is $k\Delta$.

When the coincidence $i = j$ occurs, Alice has Bob's signature because she has his signed preliminary agreement N_{B}, his signed message G, and the beacon's signed signal H. Bob's escape clause in N_{B} does not apply because $t+\Delta$ is the only time when $j = i$ occurs and the X in Alice's message does decode into $d(K)$.

If Alice follows her protocol, then she sends Bob the encrypted decoding key $X = E_{p_i}(d(K))$ only after receiving N_{B} and G from him. Thus if the beacon's signal at time $t+\Delta$ contains the decoding key $\bar{d} = d(p_i)$, then she already has Bob's receipt, as explained above.

As long as the beacon does not broadcast at time $t+\Delta$ the decoding key $\bar{d}$ for the key p_i used by Alice at time t, Bob cannot compute $D_{d(k)}(M) = E_{\mathrm{B}}(M)$. Thus, DIS cannot be computed without $d(K)$. Now, the argument in the preceding paragraphs shows that whenever the decoding key $d(K)$ is divulged, Alice must have Bob's receipt for DIS. Thus Bob cannot find DIS, or cause it to be disclosed, without his first giving Alice the receipt.

Let us now assume that at time t Alice tries to cheat Bob and let the beacon's signal at time t be $S_{\mathrm{BN}}(p_1, \ldots, p_k, d, j, t)$. If in response to Bob's message $S_{\mathrm{B}}(dis, i, t+\Delta)$ she sends $S_{\mathrm{A}}(dis, Y, i, t)$, where $Y \neq E_{p_i}(d(K))$, then even if at time $t+\Delta$ the beacon's signal (6) will satisfy $j = i$, Alice will not have Bob's receipt because of the escape clause in Bob's preliminary agreement N_{B}.

If Alice does not respond at time t with a message of the form $S_{\mathrm{A}}(dis, Y, i, t)$, then Bob's protocol realizes this before time $t+\Delta$ and terminates. Alice will then have Bob's receipt only if the beacon's signal (6) at time $t+\Delta$ satisfies $j = i$, and the probability for this is $1/k$.

Thus Alice's probability of successfully cheating is $1/k$. The probability of her being caught, resulting in the disclosure procedure (which she presumably wants to complete) being terminated by Bob, is $1 - 1/k$.

Comment. What happens if in her preliminary agreement Alice uses an M which is not $E_K(E_{\mathrm{B}}(\mathrm{DIS}))$? This is, in a sense, an artificial issue. By Alice's statement N_{A}, the string $D_{d(\mathrm{B})}(D_{d(K)}(M)) = R$ *is* her disclosure. If the disclosure procedure goes through, then Bob would be comitted to the confidentiality of this R whatever it may be. However, if R does not have the properties promised by Alice in the Disclosure Agreement (DA), for example, if she stated that DIS is a sorting program and R is a nonsense message or a description of a chemical process, then Bob is not obligated to confidentiality because the excape clause in the DA will apply. Questions arise only when it is difficult to determine if R satisfies the conditions in the DA. Thus Bob should insist that the DA define the disclosure as accurately as possible without actually revealing it. How to do this is a problem associated with disclosures, no matter what signature method is used.

The above problem does not arise in what may become the main application of the disclosure protocol, namely, disclosure of personal information. If Alice is a keeper of a data bank of personal files and agrees to disclose to Bob medical records of a person P, then DIS is uniquely defined by specifying P in the Disclosure Agreement. If for the M in Alice's N_{A} the decoding $D_{d(\mathrm{B})}(D_{d(K)}(M))$ does not read like P's medical record, then Bob is not obligated to confidentiality. If that decoding looks like P's record but is not, then there is against Alice a provable case of supplying false information.

6. Certified Mail

We can now formulate protocols for certified mail. Assume that Alice wishes to send Bob a message M in return for a receipt. There are two possible cases.

In the first case Alice does not mind Bob's seeing the message before he formally agrees to accept it. Bob wants a signed message so that Alice will not be able to disown it later. Alice wants a receipt.

This case is subsumed by contract signing. Namely, Alice and Bob formulate a contract stating that Alice is sending, and Bob is accepting, the message M. They then sign this contract, thereby completing the certified mail procedure.

If Alice does not want to show Bob the message beforehand and he is still willing to accept it, then they follow the disclosure protocol but without the disclosure agreement because no promise of confidentiality is involved.

By definition, Bob's receipt is for $D_{d(\mathrm{B})}(D_{d(K)}(M))$, whatever that may be, where M is the string contained in Alice's N_{A}.

7. Discussion and Conclusions

We want to discuss three aspects of our solution: the feasibility of a beacon, the implementation and reliability of the protocols, and the role of the judge in case of disputes.

It is clear that the viability of our protocols depends on the "honesty" of the beacon. If any participant has advance knowledge of the beacon's messages, then he can cheat his partner. Thus the beacon must be managed by a trusted party and ensured to be tamper proof. The fact that there is only one beacon with a very simple behavior makes it easier to achieve this. The beacon should use some physical method, rather than a pseudo random number generator, to produce the random numbers it needs.

If the beacon breaks down during its operation, say between time t and $t+\Delta$, the protocol's security is not impaired. As soon as the beacon resumes, the contract signing or disclosure protocols can resume.

It was pointed out by J. Reif that the beacon may be temporarily locally jammed either accidentally or maliciously. This may create a situation where one party to a transaction will know that a contract is already signed while the other, who was unable to listen, does not know. To prevent this, we can have a backup station which records and preserves the beacon's signals for a short duration. The protocol can be suspended and this station can be actively queried for a missing signal when the need arises.

Contract signing and disclosures, as well as digital signatures in general, can be implemented by rather simple special purpose signature machines. Such a machine, once given the contract or disclosure text, will execute the protocol quickly and reliably, and the whole exchange will be machine to machine without human intervention. The signature machine will also automatically prepare the preliminary agreements from the contracts or disclosures in question.

What value for k will give sufficient protection? An argument can be made that a moderate value such as $k=30$ or $k=100$ is sufficient for all practical purposes. If $k=100$ and $\Delta=3$ seconds, then an expected time of about 5 minutes will be required for the completion of a transaction.

For $k=100$, the probability of successfully cheating is $1/100$. We should bear in mind that a transaction such as contract signing is *not* an adversary game where the participants try to obtain each other's signature without giving their own signatures. If there are no safeguards, then participants may delay returning their signature and thereby gain an advantage. But when there are safeguards such as use of an intermediary, then current experience is that in the overwhelming majority of instances participants play according to the rules and do not try to cheat.

Thus, with the contract signing and disclosure protocols proposed here, we should expect that only very rarely will a participant even try to circumvent the signature box and cheat a partner to a transaction. In the rare event that somebody will try to cheat, his chances of success are, for $k=100$, just $1/100$ and his probability for being immediately found out and thereby losing the whole deal is $99/100$. Note that nowadays the percentage of signature frauds discovered immediately upon commision is much smaller. The threat of almost certain exposure will in itself act as a further deterrent. All in all, a value such as $k=100$ will be sufficient to make the occurrence of successful frauds very rare.

As a further precaution against a participant who tries a rash of cheating attempts against many partners, the network can have a central clearinghouse for complaints involving fraud attempts. A participant against whom there are many complaints, and this is bound to happen if he makes numerous attepts to cheat, can be suspended from the network.

The judge is not a trusted intermediary in disguise. When Alice and Bob carry out a transaction such as contract signing according to the protocol, each has the other's binding signature. The validity of the signatures is mathematically testable, so that there is no way they can disown the obligation. Still, it may happen that a party to an agreement chooses to cause trouble and goes to court. But these cases are a relative minority. If matters deteriorate to the point that a large proportion of all contracts end in court, then the whose notion of a contract as an instrument for preventing disputes becomes impractical. Thus under normal conditions the judge will not be somebody to whom *every* contract is brought for signature verification, but an authority for settling relatively rare disputes concerning signatures.

Since the verification of valid signatures is so unequivocal, a fine can be prescribed for anybody who brings a nuisance suit about the validity of signatures. This will not prevent lawsuits involving the substance of contracts, but will prevent litigation concerning valid signatures.

In summation, we have a convenient method for transaction protection which, by protocols not much more complicated than the basic digital signature procedures, affords a great measure of security. Currently many transactions such as instructions to brokers and fund transfers are most often done without on the spot signatures. It would be desirable to have binding proof for the authorship and receipt of such instructions, but present day paper documentation makes it impossible to keep up with the volume of traffic. In the coming era of electronic computer to computer mail, transaction protection can easily be extended to cover these situations.

We feel that protection of privacy through the disclosure protocol, which obtains a receipt from the receiver of confidential personal information, may become an important application of the methods in this paper.

REFERENCES

1. M. BLUM, "How to Exchange (Secret) Keys," Memo No. UCB/ERL/M81/90.
2. M. BLUM AND M. O. RABIN, Mail certification by randomization, in preparation.
3. W. DIFFIE AND M. E. HELLMAN, New directions in cryptography, *IEEE Trans. Inform. Theory*, **II-226** (1976), 644–654.
4. M. O. RABIN, "Digital Signatures and Public-Key Functions as Intractable as Factorization," MIT/LCS/TR-212, 1979.
5. M. O. RABIN, How to exchange secrets by oblivious transfer, manuscript.
6. R. L. RIVEST, N. SHAMIR, AND L. ADELMAN, A method for obtaining digital signatures and public-key cryptosystems, *Comm. ACM* **21** (1978), 62–65.