

Computer Science Denartment Technion-Israel Institute of Technology

| | 36 | S. Even, O. Goldre | eich, and S. Micali |
|----------------|--|---|---------------------|
| | m and his secret key such that | U (using his secret key) can quickly g | renerate a and |
| · | m and ms sector key, such that | Using its secret key) can quickly g | cherate o and |
| · | | | |
| | anyone can quickly verify the va | alidity of σ , using U's public key. However, and this segret key. We stress the | ever, it is hard |
| P | to tolge o 2 signatures without i | knowledge of his secret key. We stress the | lat signific is a |
| | | <u> </u> | Gr. Att. |
| | | | |
| - | | | |
| 7 | _ | | |
| | And the state of t | | |
| Marie Marie | | | |
| () | · · · · · | | |
| | | | |
| | | | |
| | | | |
| - | | | |
| | | | |
| | | | |
| • | | | |
| | | | |
| | | | |
| ig" | | | |
| 7 | | | |
| | | | |
| <u> </u> | | | |
| | | | |
| 21. | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| <u> </u> | | | |
| | | | |
| _ | | | |
| | | | |
| <u> </u> | | | |
| | | | |
| <u>-</u> | | | |
| | | | |
| | be signed, with one pair of keys. | | <u> </u> |
| · <u> </u> | oc signed. With offe pair of keys. | | |
| | | | |

attempt of an adversary to forge a signature of a user after getting from him signatures to messages which are randomly selected in the message space.³ (These messages are se-

lected independently of the adversary's actions.) In both cases (chosen and random mes-

sage attacks), security means the infeasibility of forging a signature to any message for

which the user has not supplied the signature (i.e., existential forgery in the terminology

of [8]).

A sufficient condition for an on-line/off-line signature scheme, as described above, to

withstand chosen message attack is that both signature schemes used in the construction

Signature Schemes

Definition 1 (Signature Schemes). A signature scheme is a triplet. (G. S. V). of prob-

abilistic polynomial-time algorithms satisfying the following conventions:

• Algorithm G is called the key generator. There is a polynomial, $k(\cdot)$, called the key

length, so that on input 1ⁿ, algorithm G outputs a pair (sk, vk) so that $sk, vk \in$

 $\{0, 1\}^{k(n)}$. The first element. sk. is called the signing key and the second element is

the (corresponding) verification key.

• Algorithm S is called the signing algorithm. There is a polynomial $\underline{m(\cdot)}$, called

the message length, so that on input a pair (sk, M), where $sk \in \{0, 1\}^{k(n)}$ and

 $M \in \{0, 1\}^{m(n)}$, algorithm S outputs a string called a signature (of message M with signing kev sk). The random variable S(sk, M) is sometimes written as $S_{ck}(M)$.

 S_{ck} answers a query $a \in \{0, 1\}^{m(n)}$ with the random variable $S_{ck}(a) = S(sk, a)$.

| - |
|------------|
| |
| |
| |
| <i>-</i> |
| , |
| |
| k succeeds |
| , <u> </u> |
| |
| |
| , |
| |
| IF |
| |
| |
| |
| |
| |
| |
| |
| |
| |
| - |
| - |
| |
| |
| |
| |
| |
| |
| |
| |

| | 3. The General Construction |
|---------|--|
| | We first define digital signature schemes with less-stringent security properties. Namely. |
| | |
| | IL . |
| <u></u> | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| · - | |
| | |
| | |
| | |
| | |
| | |
| • , | |
| | · |
| | |
| | |
| 1 | |
| | |
| | |
| | |
| | |
| | be used to sign a single message legitimately. A one-time signature scheme is secure |
| | be used to sight a single incisate regionalizate. It one time signature softene is secure |
| | |
| | |
| | |
| | against known (resp. chosen) message attack (of certain time complexity and success |
| | |
| | |

signing algorithm S with the key SK. Namely.

| 2. | <u>ae</u> | S | (vk). |
|----|-----------|------|-------------|
| Li | = | JSK! | $u\kappa$, |

The signer stores the pair of one-time keys, (vk, sk), as well as the "precomputed sig-

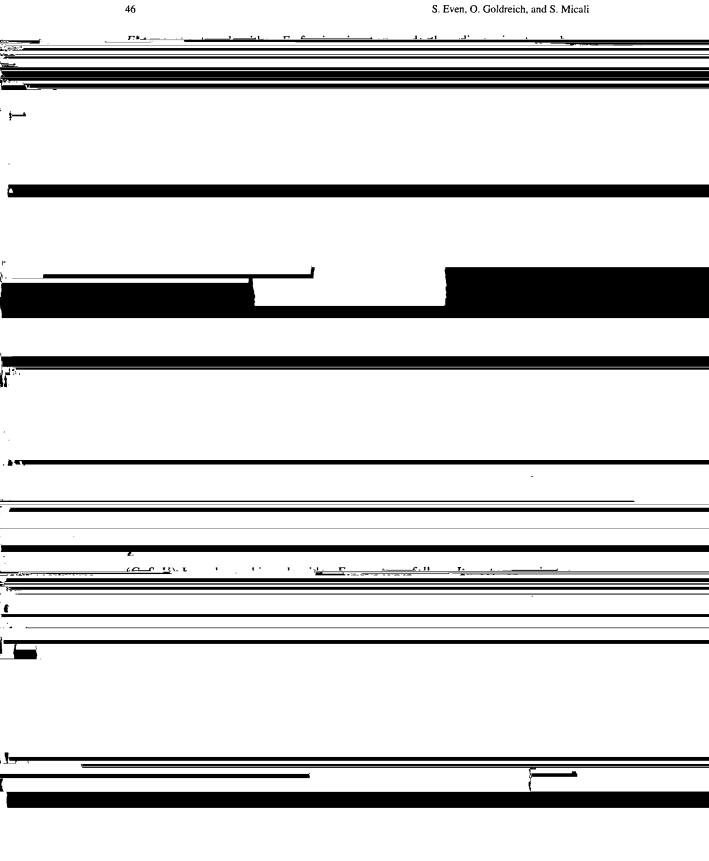
nature, Σ.

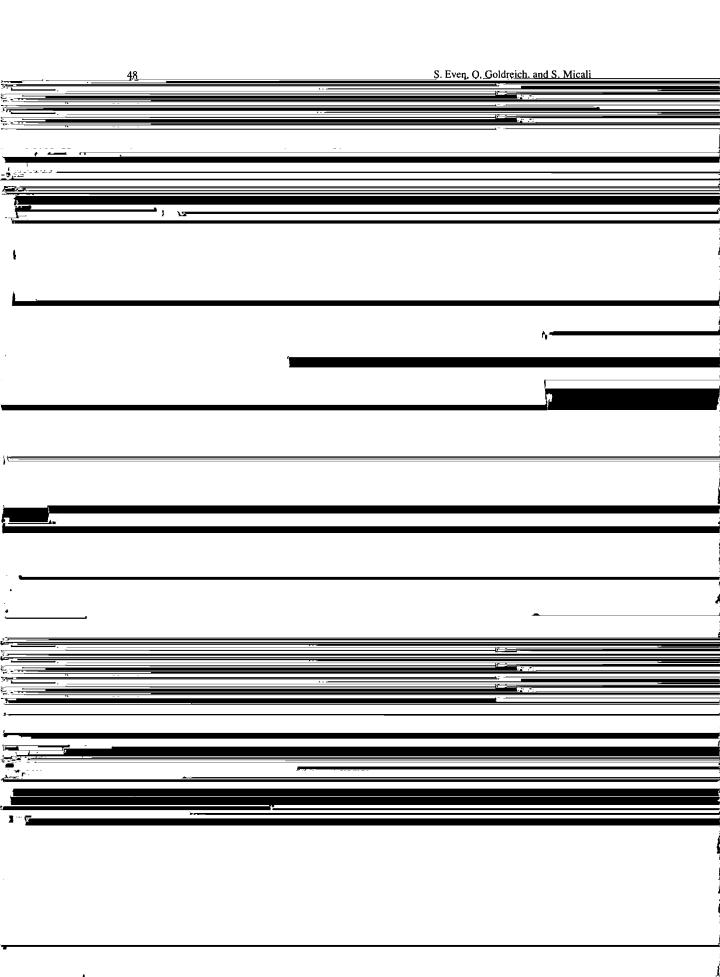
On-Line Signing

The on-line phase is performed once a message to be signed is presented. It consists of re-

trieving a precomputed unused pair of one-time keys, and using the one-time signing key

44 S. Even, O. Goldreich, and S. Micali be obtained by using collision-free hashing functions. This allows us to set $m^*(n) = n$





on which the algorithm tries to quasi-invert f. We denote $T_0 \stackrel{\text{def}}{=} (m/t) \cdot (2^t - 1)$ and

$$T_i \stackrel{\text{def}}{=} 2^i - 1$$
 for all other i's (i.e., $i = 1, \dots, (m/t)$). (T_i corresponds to the number

of times that f is iterated to form the ith component of the verification kev, where the

components are indexed by $0, 1, \ldots, (m/t)$.) On input v, supposedly taken from the

distribution
$$f^k(U_n)$$
, algorithm $A_{i,k}$ proceeds as follows. It forms a verification key as in

the key generation, except that the *i*th component is $f^{T_j-k}(v)$. That is, the verification

key is set to $y_0, y_1, \ldots, y_{m/t}$, where $y_j = f^{T_j - k}(y)$ and $y_i = f^{T_i}(x_i)$ with x_i uniformly distributed (in $\{0, 1\}^n$), for all $i \neq j$. Next, $A_{i,k}$ invokes F with this verification key,

obtaining a signature request $M = b_1 \cdots b_{m/t}$. The rest of the description is presented

We conclude that either

$$\sum_{j=1}^{m/t} \sum_{k=0}^{T_j - 1} p_j(k) \ge \frac{\varepsilon}{2}$$
 (1)

or

$$\sum_{k=1}^{T_0} p_0(k) \ge \frac{\varepsilon}{2}.$$
 (2)

Now, we consider the effect of the $p_i(b)$'s on the algorithms $A_{i,k}$. We first observe that

 $f^{T_{m/l}}(U_n^{m/l})$, where the U_n^i represent independent random variables uniformly distributed

over $\{0, 1\}^n$). For every $j \neq 0$ and $k < T_j$, we define random variables $b_1 \cdots b_{m/t}$ (resp. $c_1 \cdots c_{m/t}$) representing the message for which F has required a signature (resp. for which F has forged a signature). The probability that $A_{i,k}$ quasi-inverts on input distribution

| 4.3 | Enhancing | Security | bν | Use | of Error- | Correcting | Codes |
|-----|-----------|----------|----|-----|-----------|------------|-------|
|-----|-----------|----------|----|-----|-----------|------------|-------|

As just remarked, the security loss of a factor of m/t in the above construction is

inevitable. To avoid this loss, we need a new idea. Loosely speaking, the idea is to encode messages via a good error-correcting code and sign the encoded message rather

than the original one. This idea stands in contrast to the common practice of trying to shorten the message to be signed. Yet, the moderate increase in the length of the message

to be signed will provide a substantial benefit. The reason being that in order to forge a

Proof of Lemma 3. Let F be a probabilistic algorithm that existentially breaks the

one-time scheme, via a chosen (single) message attack, in time $T(\cdot)$ with probability $\varepsilon(\cdot)$. Hence, for every $n \in \mathbb{N}$, with probability $\varepsilon(n)$, algorithm F first asks for a signature of $M \in \mathbb{N}$. Let $\mu(M) = h \dots h$, and

 $\mu(M') = c_1 \cdots c_{m'}$. By definition of the code, $b_i \neq c_i$ for at least a ρ fraction of the *i*'s in $\{1, \ldots, m'\}$.

The inverting algorithm, A, operates as follows. On input y, algorithm A uniformly

selects $i \in \{1, ..., m'\}$ and $j \in \{0, 1\}$. Next, A forms a verification key as in the key generation, except that the (2i + i - 1)st component is y, and invokes F with this

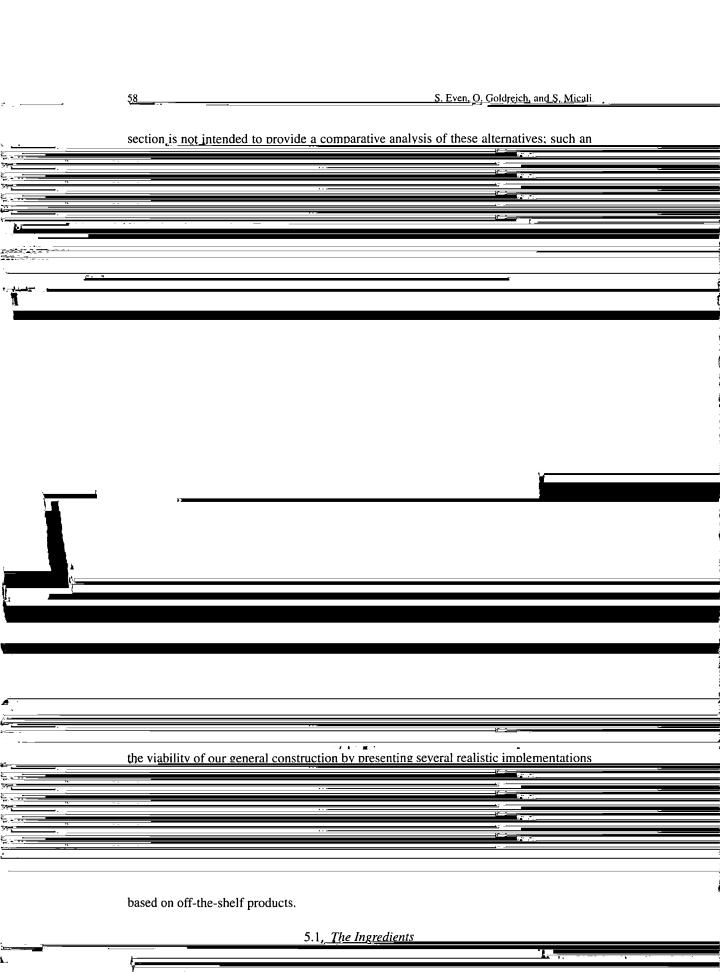
search for inverting a function, are less favorable when it is necessary to invert the

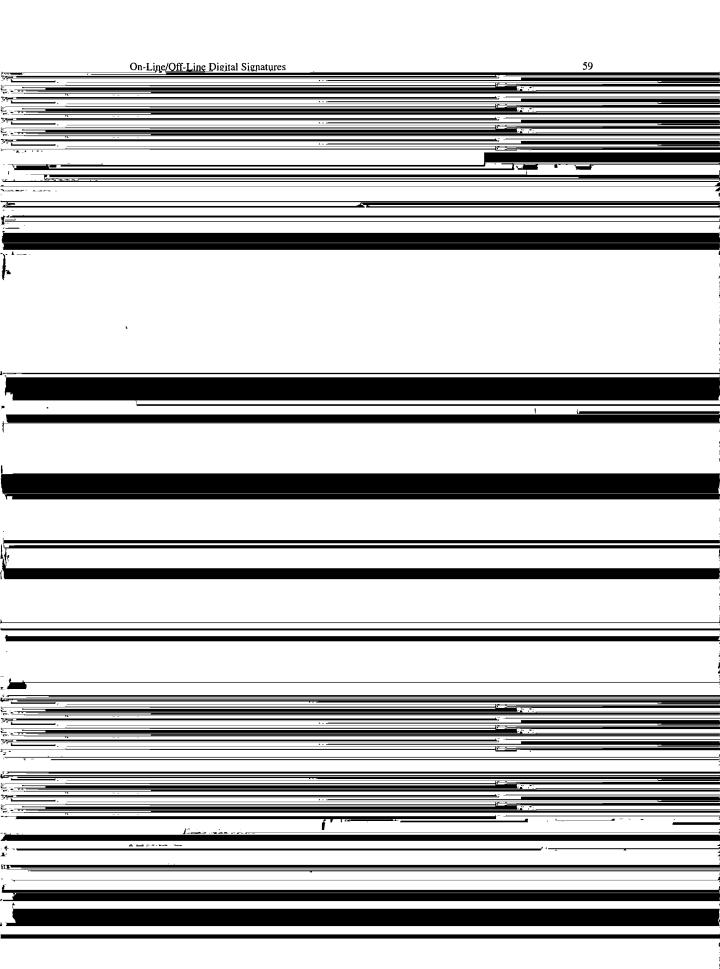
function on several instances (see Assumption 3 in the subsequent section).

Lamma 6. Cumage that T. N . N and c. N L. D are functions so that Construction?

1

can be existentially broken, via a chosen (single) message attack, in time $T(\cdot)$ with





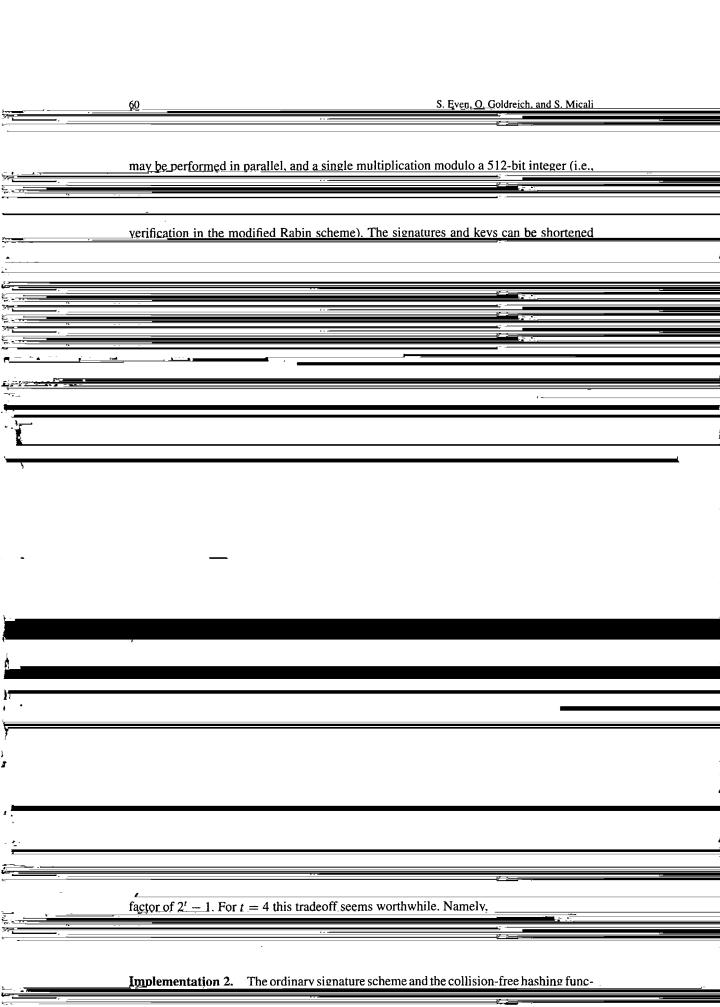


Table 1

| | | | | | | , |
|--|--------------------|-----------------|---------------------|--------------------|-------------|--------------|
| | | | | | | , |
| <u></u> | | | | | | |
| | | | | | | |
| | | | | F-1-0-12 | | ,- |
| | | | | <u>-</u> - | | , |
| и | | | | | | |
| | | ī | mplementation | | | |
| The state of the s | | | | | | |
| · | | | | <u> </u> | | - |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| <u> </u> | | | | | | , |
| | | A | | | | |
| | | | | 4.5 | | |
| <u> </u> | F.* | | | | | |
| | | | | | | |
| | | | | 'L | | |
| • | | | | | | |
| | | | | | | |
| 1 | | | | | | |
| | | | | | | |
| <u> </u> | | | | | | |
| | | | | | | |
| , | | | | | | |
| | *.= | | | | | _ |
| | | | | · · | | |
| | | | | | | |
| r | 9 | | | | | |
| | | | | | | |
| <u> </u> | | | | | | |
| <u> </u> | | | | | | |
| | | | | | | |
| • | | | | | | |
| · = | | | £ | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | Signature length 2 | <u>1</u> ,632 4 | ,1 <u>42 7,5</u> 52 | 31.037 | | |
| Transfer of the state of the st | 112 | | | | | <u></u> |
| <u> </u> | 1 | | | ₩ ===== | | - |
| | | | | | | |
| | č | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| t- | | | | | | |
| , | | | | • | | |
| | | | | | | |
| ! e | | | | | | |
| | | | | | | |
| | | | | | | |

Table 2

| Q | T | ε |
|-----------------|-----------------|-------------------|
| 104 | 10 ⁶ | 1 14,000 |
| 10 ⁴ | 10 ⁷ | $\frac{1}{1,400}$ |
| 104 | 108 | $\frac{1}{140}$ |

Implementation 1). Thus, the success probability of an attack which asks for O messages

to be signed and runs in time allowing T DES computations is bounded by

$$\frac{256 \cdot T}{D} \cdot Q$$

We stress that O is upper-bounded by the number of messages signed by a single instance

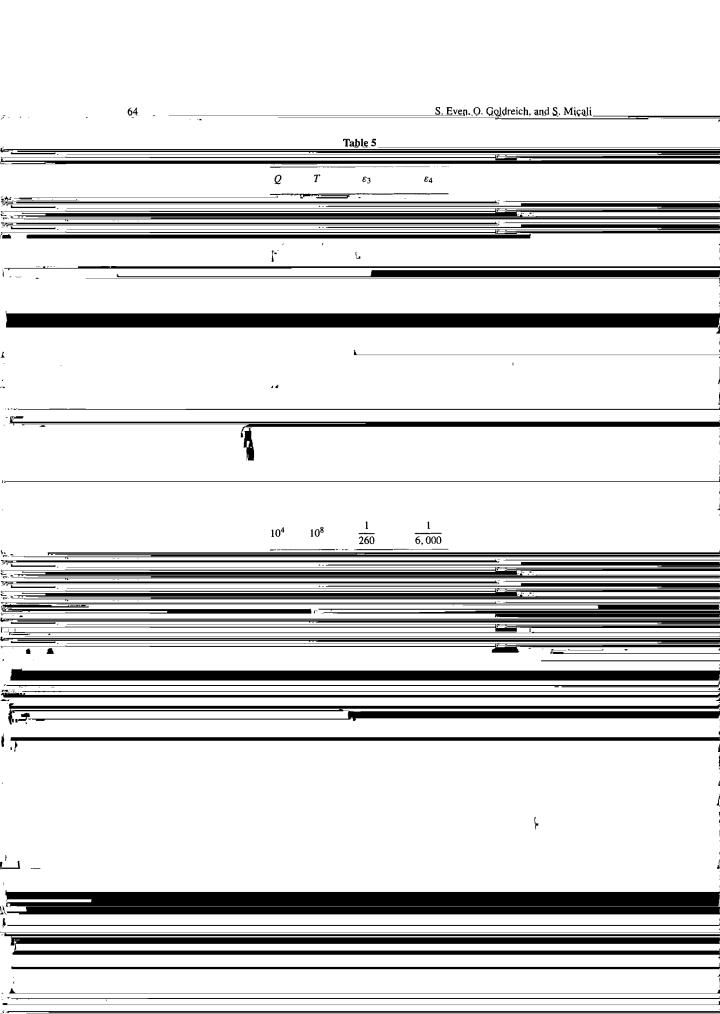
of our on-line/off-line signature scheme, throughout the "life time" of this instance. It

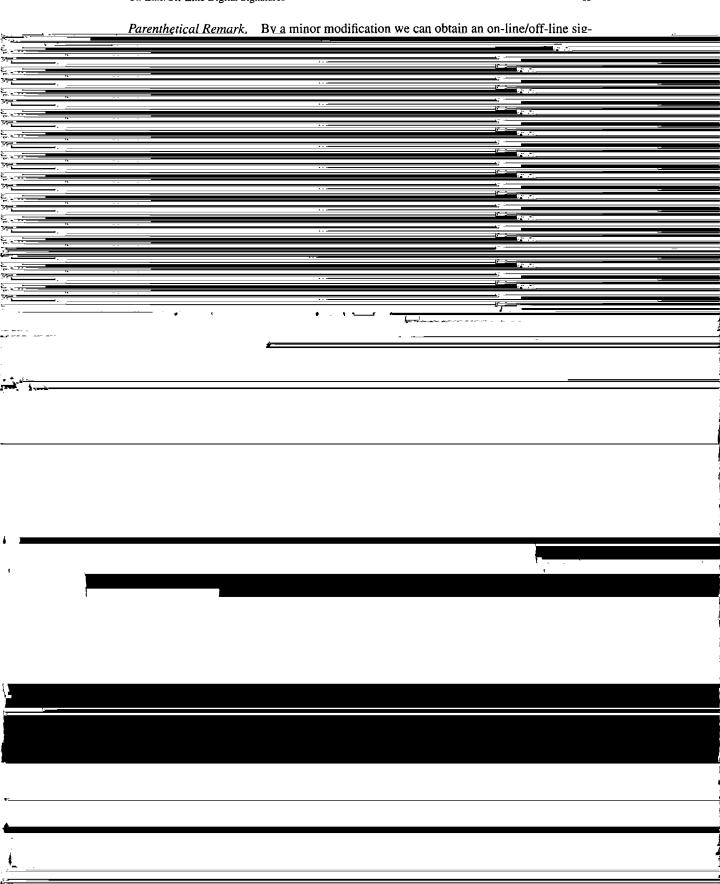
(i.e., Q) is not the total number of messages which can be signed by all instances of

our system. Recall that an instance of the signature scheme is obtained by running the

key generator. Typically, each user generates a new instance of the signature scheme

which it uses for a bounded time period. Thus, we believe that it is safe to assume that





denoted \bar{r} , is identical to a reference sequence used in a previous message. We denote this previous message by $M' = c_1 \cdots c_n$. Since $M \neq M'$, a position i exists in which

the two messages differ (i.e., $b_i \neq c_i$) and it follows that the signature M contains a

signature $S_{cv}(r_i)$ was not part of the signature obtained for M', since $c_i \neq b_i$). With very

| E | [3] Dameard, I., Collision-Free Hash Functions and Public-Kev Signature Schemes. EuroCrynt 87, LNCS. |
|--|---|
| | |
| <u>.</u> | |
| | |
| | |
| | |
| | |
| TO A STATE OF THE | |
| The second secon | |
| 1 | |
| æ. | |
| L= | |
| | |
| | |
| П. — | |
| 4 | |
| | |
| | |
| | |
| | [4] Even, S., Secure Off-Line Electronic Fund Transfer Between Nontrusting Parties, in Smart Card 2000: |
| | [4] Even. S., Secure Off-Engle electronic Fund Transfer Between Hondusting Faciles, in Smart Cara 2000. |
| \$ | |
| | |
| The second secon | |
| | The Future of IC Cards, D. Chaum and I. Schaumuller-Bichl (eds.), North-Holland, Amsterdam. 1989. |
| | |
| | |
| | pp. 57–66. |
| | [5] Even, S., Goldreich, O., and Yacobi, Y., Electronic Wallet, Advances in Cryptology: Proc. Crypto 83, |
| | D. Chaum (ed.), Plenum, New York, 1984, pp. 383–386. [6] Even, S., Goldreich, O., and Micali, S., On-Line/Off-Line Digital Signatures, Advances in Cryptology: |
| | |
| | |
| | |
| · <u></u> | |
| <u>-</u> | |
| · · · · · · · · · · · · · · · · · · · | Proc. Crypto 89. G. Brassard (ed.), LNCS. Vol. 435. Springer-Verlag, Berlin. 1990, pp. 263-277. |
| | |

[7] Goldreich, O., Two Remarks Concerning the Goldwasser-Micali-Rivest Signature Scheme. Advances in