Better than BiBa: Short One-time Signatures with Fast Signing and Verifying

Leonid Reyzin & Natan Reyzin

CSC 774: Network Security

Spring 2003

Amit S Gambhir

Introduction

- **Problem:** To build short one-time signatures with fast signing and verifying capabilities
- **Motivation:** BiBa one-time signature [Per01] scheme has very fast signature verification but comparatively slower signing and bigger signatures.

[Per01] Adrian Perrig, "The BiBa One-Time Signature and Broadcast Authentication Protocol," in Proceedings of the ACM Conference on Computer and Communications Security, November, 2001.

Outline

- Prior Work
  - One time signature schemes
  - BiBa Signature and its disadvantages
  - Applications of One-time Signatures
- The proposed scheme
  - Recommended Algorithms
  - Efficient Modifications
  - Comparison with BiBa
- Future Work

One time Signatures

- One time public and private keys
- Each key pair allows for signing (and verifying) of only one message.
- Verification time is usually very less
- Good for broadcast environments, where quick source authentication is of utmost importance. e.g. Wireless n/ws

The BiBa Signature Scheme

<Just discussed by Rich>

Made for the Broadcast Authentication Protocol

+ One time signature
+ Fastest verification
  - Small communication overhead for receiver
  - Instant authentication (no buffering and packet loss)
+ Short signatures
– Longest signing time

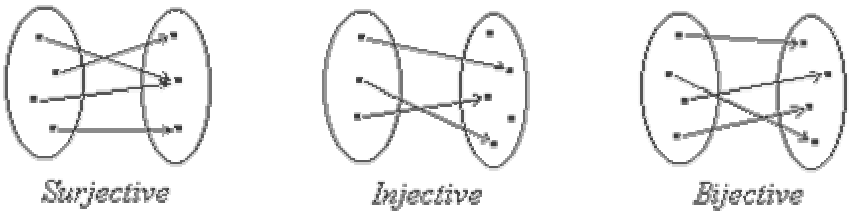
Applications of One-time Signatures

1. Regular signature schemes
2. On-line/Off-line signature schemes
3. Forward-secure signature schemes
4. Multicast packet authentication
5. BiBa Broadcast authentication scheme

The Proposed Scheme: Preliminaries

- Parameters:
  b: # of bits of the message as input to signature scheme
  t & *k*: choose such that $C^t_k > 2^b$
- Lets define a set of real numbers:
  $T = \{0, 1, 2, \ldots, t\text{-}1\}$
- Let *f* be a one-way function (central idea)
- And *S* be a 'bijective' (later) function:
  Input: message *m*
  Output: *m*-th k-element subset of the set *T*

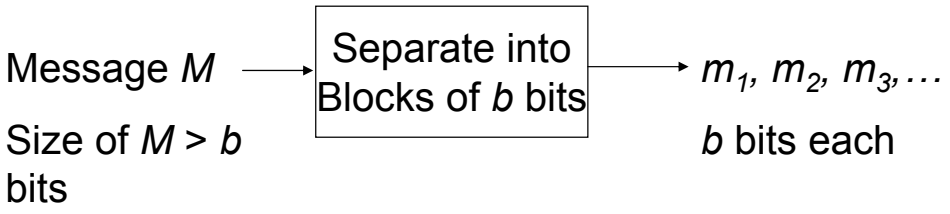
Bijective functions



- Each element of input is mapped onto one and only one element in output
- Each element of output is mapped onto one and only one element in input.
- Essentially, you can pair all elements in input and output with each other.
- Intuitively, both the sets must be of equal size.

The proposed scheme

Step # 1:



Key Pair Generation

- **Step # 2: Generate key pair**
  - Generate *t* number of random *l*-bit values
  - Let these be the secret key: SK = $(s_1,\ldots,s_t)$
  - Compute the public key: PK = $(v_1,\ldots,v_t)$ where: $v_i = f(s_i)$
  - Where *f*() is the one-way function

Signature Generation

- **Step # 3: Generation of signature**
  - Take each 'mini' message m_i and interpret it as an integer value between 0 and $2^b - 1$
    - Possible because each m_i is b-bits long
  - Feed each integer to input of bijective function S
  - S will output the m-th k-element subset of T: $\{i_1, \ldots, i_k\}$
  - From SK = $(s_1, \ldots, s_t)$, reveal corresponding values $s_{i1}, \ldots, s_{ik}$ as signature

Signature Verification

- Let $(s_1', s_2', \ldots, s_k')$ be the received signature on a message m
- Interpret m as an integer between 0 & $2^b - 1$
- Using S, find the m-th k-element subset: $\{i_1, \ldots, i_k\}$
- Verify that $f(s_1') = v_1$ and so on, where: $(v_1, \ldots, v_t)$ was the public key.

Efficiency Analysis

- Key generation:
  - Requires t evaluations of the one-way function
  - Secret key size = lt bits
  - Public key size = $f_l t$ bits
    - f_l = length of the one-way function output
- Signature generation: Time required to find m-th k-element subset of T
- Verification: same time as signing + k evaluations of the one-way function

Significance of Parameters

- Public Key, PK = $(v_1,\ldots,v_t)$. Hence size of PK is directly proportional to t.
- Signature size and verification time depend on k.
- Also, $C^t_k > 2^b$ is required to generate signatures on b-bit messages.
- Mathematically, multiple choices for t and k are available

Security of the scheme

- Bijective function *S*:
  - REM: each input corresponds to one and only one output & vice-versa
- Each *b*-bit message *m* corresponds to a different *k*-element subset of *T* because of the combination formula.
- The one-way function *f* is infeasible to invert.

Signature forging

- Objective: to forge a signature on a new message after an adaptive chosen message attack.
- The forger already has a part of the secret key, i.e. *k*-bits SK' = $\{s_{i1},\ldots,s_{ik}\}$
- The forger would need to invert the one-way function *f* on at least $t - k$ values in the public key for which the secret key has not been revealed.

Proposed Algorithms for S

Algorithm #1: $C^t_k = C^{t-1}_{k-1} + C^{t-1}_k$

- If the last element of T belongs to the subset, then t-1 elements remaining from which k-1 need to be chosen; otherwise, t-1 remaining from which k need to be chosen
- Input: (*m,t,k*)
- *If* $m < C^{t-1}_{k-1}$
  - then add t-1 to o/p subset and recur on (*m,k-1,t-1*)
  - Else add nothing to o/p subset and recur on (m,C^{t-1}_kk, t-1)
- C^t_k can be pre-computed. Each recursion requires one division and one multiplication of O(k log t) – bit number to compute C^{t-1}_{k-1} and C^{t-1}_k i.e. cost = $k\log^2 t$
- For t levels, the total cost = $t\ k\log^2 t$

Proposed Algorithms for S

Algorithm # 2: If k elements are selected from *T*, then, for some *i*, *i* elements come from the first half of *T* and *k-i* elements from the second half.

- The two halves (or subsets) recur and their o/ps are combined.
- Final cost = $O(k^2 \log t \log k)$
- Practical implementation: t = 1024, k = 16 and random *m* takes 0.09 ms on 1700Mhz P4

Subset-Intractable functions

- Replace bijective *S* w/ a cryptographic hash function *H* (like SHA-1): Relax constraints to improve efficiency
  - No guarantee that 2 messages result in distinct *k*-element subsets. Only “infeasible”
  - Subset of T may contain at most *k* elements.
  - Infeasible to find two messages m_1 and m_2 such that: H(m_2) is a subset of H(m_1).

Expected results of the scheme

- Efficient signing: one evaluation of *H*.
- Efficient verifying: one evaluation of *H* and *k* evaluations of the one-way function *f*.

Comparison with BiBa

	BiBa	New Scheme
Signing	$2t$ calls to the random oracle	One call to H
Verifying	k calls to the random oracle	One call to H
Values of t and k for same security level	Larger	smaller

Security in numbers

- Probability that after querying H on a single message m, the adversary is able to forge a signature on m w/o inverting the one-way function f is $(rk/t)^k$,
  - r = number of messages the adversary already has the signatures for
- "# of bits" of security = $k(\log t - \log k - \log r)$
- For k = 16, t = 1024 and r = 4, security level = 2^{-64}
- For BiBa scheme, it is 2^{-58}

Future Work

- Incomplete
  - No name!
  - Requirements for *H* need to be formalized
  - The signing, verifying and security need to be tested for 'many' hash functions for practicality.
- Relies on the assumption that one-way functions exist.
  - Need to find out if practical solutions like trapdoor functions are viable and provide the same, if not better, level of security.

Thank You

- Questions?