Sequential Aggregate Signatures from Trapdoor Permutations

Anna Lysyanskaya, Leonid Reyzin, Silvio Micali, and Hovav Shacham

Non-sequential Aggregates [BGLS03]

$$
\begin{tikzcd}[row sep=small]
 & & \text{Sign} & & \text{Aggregate} \\
u_1: & M_1 \arrow[rr] & & \sigma_1 \arrow[dr] & \\
u_2: & M_2 \arrow[rr] & & \sigma_2 \arrow[r] & \sigma \\
\vdots & \vdots & & \vdots & \\
u_n: & M_n \arrow[rr] & & \sigma_n \arrow[uur] &
\end{tikzcd}
$$

- Related to BLS short signatures [BLS01]
- Instantiated using bilinear map

Aggregate Signatures [BGLS03]

- A single short aggregate provides nonrepudiation for many different messages under many different keys
- More general than multisignatures
- Applications:
  - X.509 certificate chains
  - Secure BGP route attestations
  - PGP web of trust

Verisign

Versign Europe

NatWest

NatWest WWW

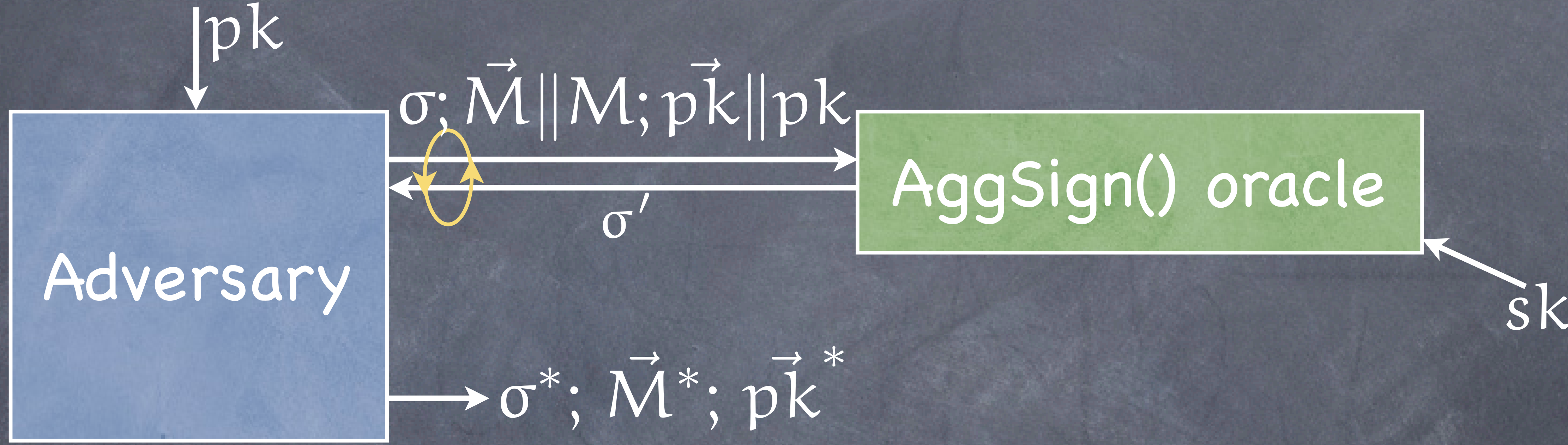
Sequential Aggregates

Sign and Aggregate

$$
\begin{tikzcd}
U_1 : & M_1 \arrow[r] & \sigma_1 \arrow[dr] & & \\
U_2 : & M_2 \arrow[rr] & & \sigma_2 \arrow[ddr, "\ddots"] & \\
\vdots & \vdots & & & \\
U_n : & M_n \arrow[rrr] & & & \sigma_n
\end{tikzcd}
$$

- Signing and Aggregation are a single operation
- Inherently sequenced; not appropriate for PGP
- Can be instantiated using RSA

Sequential Aggregate Chosen-Key Model



Nontriviality:

- σ^* is a valid sequential aggregate
- challenge key pk = pk_j^* for some j;
- No oracle query at $pk_1^*,\ldots,pk_j^*;M_1^*,\ldots,M_j^*$.

Trapdoor Permutations

- A permutation family Π over D:
  - Generate: $(s,t) \leftarrow$ Gen
  - Evaluate: $\pi(\cdot) = \mathrm{Eval}(s,\cdot)$: $D \rightarrow D$
  - Invert: $\pi^{-1}(\cdot) = \mathrm{Invert}(s,t,\cdot)$: $D \rightarrow D$
- Here, D is a group over some operation $*$.

Trapdoor Permutation Features

- One-way: hard to invert without trapdoor t.
- Homomorphic: each π is a permutation over some group operation $\times$ (not necessarily the same as $*$)
- Claw-free [GMR88]: hard to find claw (x,y) s.t. $\pi(x)=g(y)$ (where g is an additional permutation of D)
- Certified [BY96]: easy to tell whether a given s corresponds to a valid permutation (s,t).

Full-domain Hash Signatures [BR93,C00]

- Use random oracle hash H: $\{0,1\}^* \to D$
- Signature scheme:
  - Key Generation: (PK,SK) = (s,t) ← Gen
  - Sign $M \in \{0,1\}^*$:
    $h \leftarrow H(M) \in D$; $\sigma \leftarrow \mathrm{Invert}(s,t,h) \in D$
  - Verify σ: $h \leftarrow H(M) \in D$; check $\mathrm{Eval}(s,\sigma) = h$.
- Secure if Π is one way;
  better reduction if Π is homomorphic.

Trapdoor Sequential Aggregate Signatures

- Key gen for each user: $(s,t) \leftarrow \text{Gen}$
- Aggregate Sign M under (s,t), along with σ on $M_1, \ldots, M_i$ under $s_1, \ldots, s_i$:

  verify that σ is valid;

  $h \leftarrow H(M_1, \ldots, M_i, M, s_1, \ldots, s_i, s)$;

  $\sigma' \leftarrow \text{Invert}(s, t, h * \sigma)$.

- Verify σ on $M_1, \ldots, M_i$ under $s_1, \ldots, s_i$:

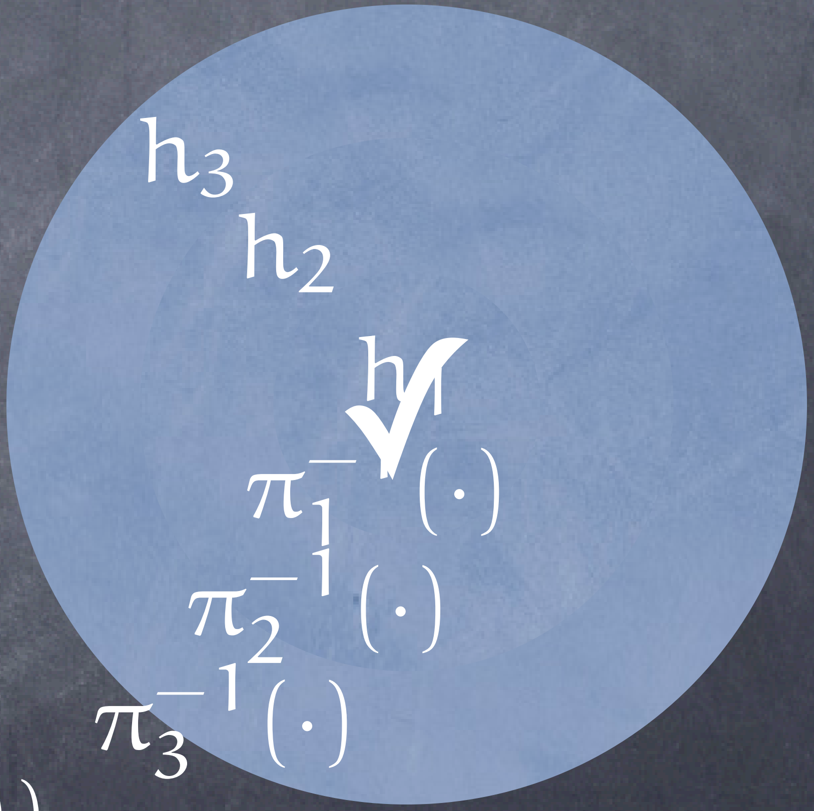
  for $j = i, \ldots, 1$ do:

  $\sigma_{j-1} \leftarrow \text{Eval}(s_j, \sigma_j) * H(M_1, \ldots, M_j, s_1, \ldots, s_j)^{-1}$

  accept if $1 = \sigma_1$.

An Example

- Let $h_i = H(M_1,\ldots,M_i,s_1,\ldots,s_i)$ for each i
- Then:



$$
\begin{aligned}
\sigma_1 &= \pi_1^{-1}(h_1) \\
\sigma_2 &= \pi_2^{-1}(h_2 \cdot \pi_1^{-1}(h_1)) \\
\sigma_3 &= \pi_3^{-1}(h_3 \cdot \pi_2^{-1}(h_2 \cdot \pi_1^{-1}(h_1)))
\end{aligned}
$$

Trapdoor Aggregate Signature Security

- Theorem: Secure (in random-oracle model) against existential forgery in the sequential aggregate chosen-key model if Π is a certified, one-way permutation family.
- Theorem: Better reduction if Π is claw-free.

Instantiating With RSA

- Each user has $N=pq$, along with $e \cdot d = 1\ (\varphi(N))$
- Pub key (N,e), priv key (N,d); $\pi(x) = x^e$, $\pi^{-1}(x) = x^d$.
- Problems:
  - domain is Z_N^*, not Z_N;
  - RSA not certified: can't tell if (N,e) well-formed;
  - N is different for each user.
- Not just a "proof problem"!

Certifying RSA

- Extend $\pi(\cdot)$ to Z_N :
  - define $\pi(x) = x$ when $\gcd(x,N) \neq 1$
  - Use + as group operation: $\sigma' \leftarrow (h+\sigma)^d$
    ($\times$ is still used in security proof)
- Certify (N,e):
  - require e > N and prime,
    so $\gcd(e, \varphi(N)) = 1$. [CMS99]

Dealing with Ns

It can happen that $\sigma_i > N_{i+1}$. Two solutions:

- Require $N_1 < N_2 < \ldots < N_n$.
- Require that each N_i be k bits long; output overflow bit 1 when $\sigma_i > N_{i+1}$, 0 otherwise (aggregate grows by one bit per signature).
  - This generalizes: if keys are within 2^t factor of each other, output t extra bits per aggregation step.

Conclusions

- An aggregate signature provides nonrepudiation on many messages by many keys
- Sequential aggregates are inherently sequenced; signing and aggregation are a single operation
- Can instantiate using RSA;
  requires making RSA a certified permutation