

Chameleon Hash Functions & One-time Signatures

Payman Mohassel

University of Calgary

One-Time Signatures (OTS)

$$OTS = (KG, sign, verify)$$

$$(vk, sk) \leftarrow KG$$



Signer

$$\longleftarrow m$$

$$\xrightarrow{\sigma = sign(m, sk)}$$

$$\longleftarrow m', \sigma'$$

$$verify(vk, m', \sigma')? = 1$$

vk



Forger

Strongly unforgeable if $m' = m$ is allowed

Applications of OTS

- Tree-based constructions
 - OTS => standard signatures, [GMR' 88, ...]
- Authenticate messages in various networks
 - Multicast/broadcast networks, [Perrig' 01, ...]
 - Sensor networks, [DK' 09, ...]
- Building block for other constructions
 - Online/offline signatures, [EGM' 96]
 - IBE => IND-CCA PKE, [CHK' 04]
 - *standard* signatures => *strong* signatures, [HWZ' 07]
 - ...

General Constructions for OTS

- From one-way functions
 - [Lamport' 79, ...]
 - Signatures are long!
- Claw-free permutations
 - [GMR' 88]
 - Computationally expensive!
- Fiat-Shamir transform
 - ID schemes => standard signatures
 - Efficient and short
 - Random oracle model
- [Bellare and Shoup, 2007]
 - ID schemes => OTS
 - Standard model

OTS from Standard Assumptions

- DL-based schemes
 - [HP' 92, Groth' 06, ...]
- Factoring-based schemes
 - [BPW' 90, PP' 97, ...]
 - Fail-stop signatures
 - Not the standard RSA integer factoring assumption
- Lattice-based schemes
 - [LM' 08, ...]

Chameleon Hash Functions (CHF)

[KR00]

$$H = (Gen, h, h^{-1})$$

$$(ek, td) \leftarrow Gen$$

- Evaluation: $h(ek, m, r)$ (**randomized hashing**)

- Collision resistance:

$$(m, r) \neq (m', r') \leftarrow A(ek) \quad \text{s.t.} \quad h(ek, m, r) = h(ek, m', r')$$

- Chameleon property: (**trapdoor collisions**)

For any m, m', r

$$r' \leftarrow h^{-1}(td, m, r, m') \quad \text{s.t.} \quad h(ek, m, r) = h(ek, m', r')$$

- Similar to trapdoor commitments, [BCC' 88, ...]

Example of CHF

- A group G of order p , generator g_1
- $ek = (G, p, g_1, g_2 = g_1^x)$, $td = x$
- $h(ek, m, r) = g_1^m g_2^r$
- Finding collision \Rightarrow recover x
- Given m, m', r, x
 - $r' = (m + rx - m') x^{-1}$
 - $g_1^m g_2^r = g_1^{m'} g_2^{r'}$

CHF and OTS

- Used in similar applications
 - Online/offline signatures
 - *Standard to strong* signatures
 - ...
- [Groth' 06]
 - From trapdoor Pedersen commitment
 - Not a general construction

Our Result

- CHF \Rightarrow strong OTS
 - Simple, general
- Implications
 - Unify some previous works
 - ID schemes \Rightarrow strong OTS
 - New instantiations of OTS

CHF \Rightarrow Semi-OTS

$$H = (Gen, h, h^{-1})$$

$$OTS = (KG, sign, verify)$$

- KG:
 1. $(ek, td) \leftarrow Gen$
 2. $vk = [ek, z = h(ek, 0, r)]$, $sk = [td, r]$
- Sign m :
 1. $\sigma = h^{-1}(td, 0, r, m)$
- Verify:
 1. Accept iff $h(ek, m, \sigma) = z$

Security: signature query fixed before generating vk

Proof

- If **A** forges a signature
 - **B** finds a collision for (h, ek)
- When **A** makes a signature query for m
 - **B** computes $z = h(ek, m, r)$ for random r
 - Returns $vk = (ek, z)$ and signature $sig = r$
- When **A** forges (m', sig')
 - **B** outputs (m, sig) and (m', sig') as collision for h

Semi-OTS \Rightarrow Full-OTS

$OTS = (KG, Sign, Verfy) \Rightarrow OTS' = (KG', Sign', Verfy')$

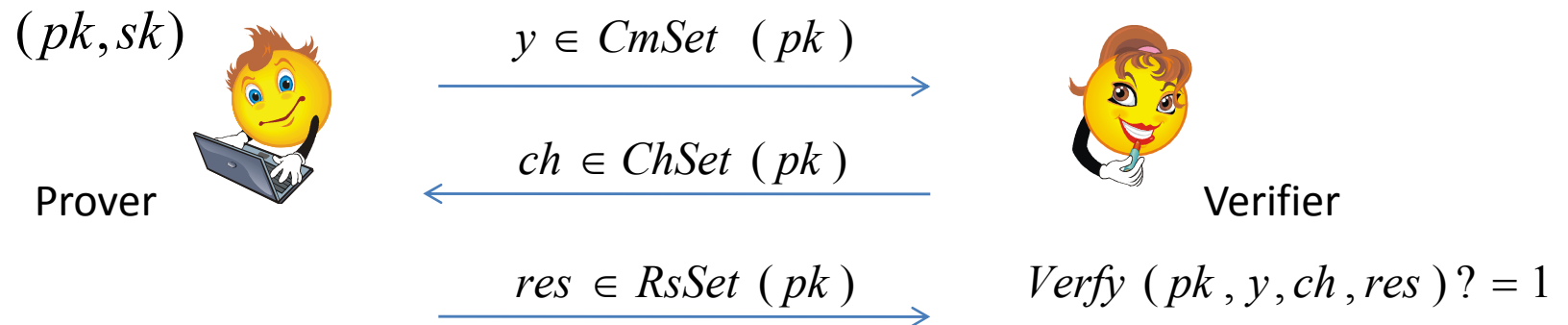
- 1) Hash the message using CHF, and randomness r
- 2) Sign the hash; output the signature and r

- KG' :
 1. $(vk_0, sk_0) \leftarrow KG$; $(ek_1, td_1) \leftarrow Gen$
 2. $z = h(ek_1, 0, r_1)$
 3. $vk = [vk_0, ek_1]$, $sk = [sk_0, td_1, r_1, z]$
- $Sign'$:
 1. ~~$\sigma_0 = sign(sk_0, m)$~~ , $\sigma_1 = h^{-1}(td_1, 0, r_1, m)$
 - $\sigma_0 = sign(sk_0, z)$
- $Verfy'$ (m, σ_0, σ_1) :
 1. Accept iff $Verfy(vk_0, h(ek_1, m, \sigma_1), \sigma_0)$

CHF \Rightarrow Full OTS

- **KG'**:
 1. $(ek_i, td_i) \leftarrow Gen$ for $i \in \{0,1\}$
 2. $z_0 = h(ek_0, 0, r_0), z_1 = h(ek_1, 0, r_1)$
 3. $vk = [ek_0, ek_1, z_0], sk = [td_0, td_1, r_0, r_1, z_1]$
- **Sign'**:
 1. $\sigma_0 = h^{-1}(td_0, 0, r_0, z_1), \sigma_1 = h^{-1}(td_1, 0, r_1, m)$
- **Verify'**:
 1. Accept iff $h(ek_0, h(ek_1, m, \sigma_1), \sigma_0) = z_0$

OTS from Sigma Protocols



Strong special soundness:

(y, ch, res) and (y, ch', res') where $(ch, res) \neq (ch', res')$

Strong honest verifier zero-Knowledge:

ZK-simulator is deterministic and of special form

ID schemes to Chameleon Hash (standard model)

[Bellare-Ristov, 08]

OTS from Sigma Protocols

- ID schemes => Chameleon Hash => strong OTS
 - Factoring, RSA, DL, ...
- [BS' 07]
 - ID scheme secure against *concurrent attacks*
 - Appears to be a stronger property
 - Based on the one-more RSA
 - Not based on ``RSA'' or ``Factoring''

Instantiations

- Via existing CHF or ID schemes
 - Factoring, RSA, DL, lattice-based, ...
- Not all are interesting!
 - But some have interesting properties

CHF from Factoring [ST01]

SETUP :

1. Safe primes $p, q \in \{0,1\}^{k/2}$, and $N = pq$
2. Random element $g \in Z_N^*$
of order $\lambda(N) = (p-1)(q-1)/2$
3. $pk = (N, g)$ and $sk = (p, q)$

EVALUATION:

On $m \in Z_N$ and $r \in Z_{\lambda(N)}$
return $g^{m||r} \bmod N$

INVERSION:

On $m, m' \in Z_N$ and $r \in Z_{\lambda(n)}$
return $r' = 2^k(m-m') + r \bmod \lambda(n)$

- Previous Works
 - Fail-stop signatures
 - p, q are of special form
- Signing is fast!

CHF from Lattices

- [Pei' 09]
 - Chameleon Hash functions based on standard lattice-based assumptions
 - Preimage samplable trapdoor functions
 - SIS, SIVP, ...

	Key Size	Signature Size	Ideal/Standard
[LM08]	$\tilde{O}(k)$	$\tilde{O}(k)$	Ideal
[CHKP10]	$\tilde{O}(k^3)$	$\tilde{O}(k^2)$	Standard
[Boyen10]	$\tilde{O}(k^3)$	$\tilde{O}(k)$	Standard
Ours	$\tilde{O}(k^2)$	$\tilde{O}(k)$	Standard

Summary

- CHF \Rightarrow Strong OTS
 - ID Schemes \Rightarrow OTS
 - New instantiations for OTS
 - Unifying some previous works
- Question
 - Better understanding of hardness of CHF
 - Can it be based on CRHFs?

Thank You!