

	Based on number-theoretic problems	VSH	Dakota
Number-th	eoretic, difficult problem	ns	
Fact	oring:		
	given $N = pq$ , fin	d $p$ and $q$ ,	
wh	ere <i>p</i> , <i>q</i> big, (odd) prime numl	pers, $p  eq q$	
Reco	ommended that $N \ge 2^{1024}$ for	high level of security	
A 10	24-bit <i>N</i> :		
1350	6641086599522334960321627	88059699388814756056670	)
	4485143851526510604859533		
	2821644715513736804197039		-
	4102086438320211037295872		-
	8187510676594629205563685 5339061097505443349998111		(

1 Introduction **2** Based on number-theoretic problems 3 VSH **4 DAKOTA** 2 / 30

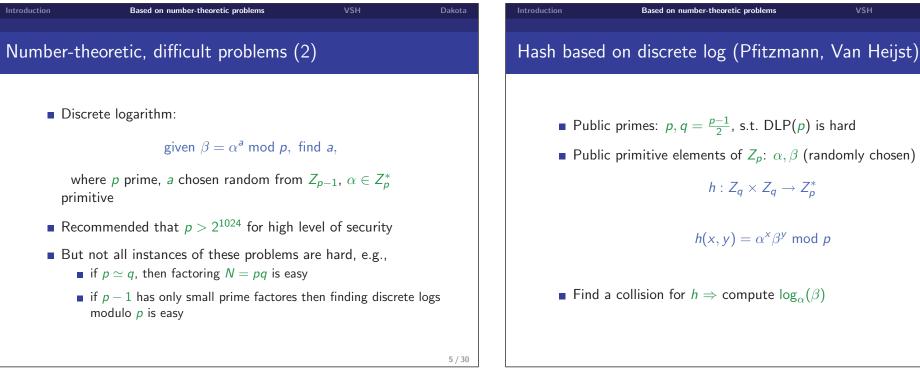
# Generic attacks

Introduction

For  $H : \{0,1\}^* \to \{0,1\}^n$  and  $h : \{0,1\}^m \to \{0,1\}^n$ , m > n

attack	rough complexity
collisions 2nd preimages	$\frac{\sqrt{2^n}}{2^n} = 2^{n/2}$
preimage	2 <sup>n</sup>

Goal: generic attacks are best (known) attacks



Hash based on factoring (Shamir)
N = pq, p ≠ q, large odd primes, α fixed, large order mod N.
Public: N, α

 $H:\{0,1\}^*\to Z^*_N$ 

$$H(x) = \alpha^x \mod N$$

• Collision:  $H(x) = H(x') \Rightarrow x - x' = k\phi(N)$ .

Based on number-theoretic problems

• With N = pq and  $\phi(N) = (p-1)(q-1)$  easy to find p and q

 Introduction
 Based on number-theoretic problems
 VSH
 Dalota

 **Mumber-theoretic hash functions** 

 • most schemes slow, e.g., no real speed-up for use in digital signature schemes

 • some schemes have unfortunate algebraic properties (may interact badly with other public-key algorithms)

 • open problem to devise efficient "provably" secure hash function

Dakota

7/30

#### Introduction Based on number-theoretic problems Dakota Based on number-theoretic problems Introduction Example. Dixon's algorithm with n = 4189Newer constructions $\sqrt{4189} \simeq 64.7$ • Use the factor base $\mathcal{B} = \{-1, 2, 3, 5, 7, 11, 13\}$ VSH - Very Smooth Hash Contini, Lenstra, Steinfeld, 2005 $x_i^2 \mod n$ factorisation of $x_i^2$ ai • collision $\Rightarrow$ nontrivial modular square roots of very smooth $58^2 \mod n = -1 \cdot 3 \cdot 5^2 \cdot 11$ (1, 0, 1, 0, 0, 1, 0)numbers modulo N (composite) $61^2 \mod n = -1 \cdot 2^2 \cdot 3^2 \cdot 13$ (1, 0, 0, 0, 0, 0, 1) efficient collision finder implies fast factoring algorithm $67^2 \mod n = 2^2 \cdot 3 \cdot 5^2$ (0, 0, 1, 0, 0, 0, 0) $69^2 \mod n = 2^2 \cdot 11 \cdot 13$ I ASH - A Lattice Based Hash Function (0, 0, 0, 0, 0, 1, 1) $74^2 \mod n = 3^2 \cdot 11 \cdot 13$ Bentahar, Page, Saarinen, Silverman, Smart 2006 (0, 0, 0, 0, 0, 1, 1)based on the problem of finding small vectors in lattices ■ $gcd(58 \cdot 61 \cdot 67 \cdot 69 + ((2^3) \cdot (3^2) \cdot (5^2) \cdot 11 \cdot 13), n) = n;$ ■ $gcd(58 \cdot 61 \cdot 67 \cdot 74 + ((2^2) \cdot (3^3) \cdot (5^2) \cdot 11 \cdot 13), n) = 59;$ 9/30

Dakota

Introduction

# Factoring - Equal squares

• Let n = pq,  $p \neq q$ , odd primes. Let  $x \in Z_n^*$ 

Based on number-theoretic problems

■ 4 square roots of x<sup>2</sup> mod n are x, -x, y, -y mod n where x ≠ ±y mod n

mod <i>n</i>	mod <i>p</i>	mod q
x	Z	W
-x	- <i>z</i>	-w
у	Ζ	-w
- <i>y</i>	- <i>z</i>	W

- gcd(x+y,n) = q, gcd(x-y,n) = p,
- Find (random) a, b s.t.  $a^2 = b^2 \mod n$ , factor n with prob.  $\frac{1}{2}$

10 / 30

# Quadratic and number field sieves

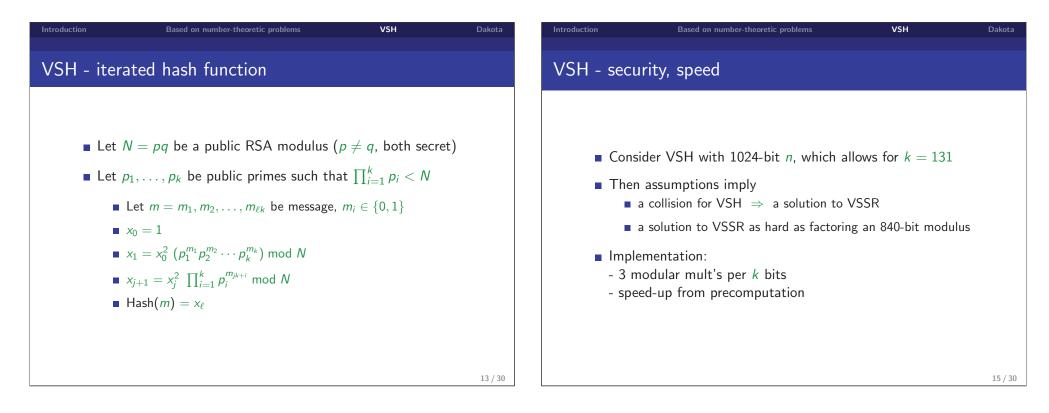
Based on number-theoretic problems

- Quadratic sieve, advanced variant of Dixon's algorithm
- Number field sieve(NFS), advanced variant of quadratic sieve
- NFS currently best known algorithm for factoring large RSA moduli
- Size of factor base:  $e^{(0.96+\mathcal{O}(1))(\ln n)^{1/3}(\ln \ln n)^{2/3}}$
- Running time:  $e^{(1.923+\mathcal{O}(1))(\ln n)^{1/3}(\ln \ln n)^{2/3}}$
- Notation:  $L[n, \alpha] = e^{(\alpha + \mathcal{O}(1))(\ln n)^{1/3}(\ln \ln n)^{2/3}}$

12/30

11/30

Dakota



Security of VSH

**VSSR Problem.** Let n = pq be a public RSA modulus (p, q secret). Let  $k \leq (\log n)^c$ . Find  $x \in Z_n^*$  such

$$x^2 \equiv \prod_{i=0}^k p_i^{\mathbf{e}_i} \bmod n,$$

VSH

where at least one  $e_i$  is odd.

VSSR Assumption: The VSSR Problem is hard.

## Computational VSSR Assumption:

Solving VSSR for n is as hard as to factor an S-bit modulus, where S is least positive integer satisfying

$$L[2^{S}, 1.923] \ge \frac{L[n, 1.923]}{k}$$

 $L[n, \alpha] = e^{(\alpha + \mathcal{O}(1))(\ln n)^{1/3}(\ln \ln n)^{2/3}}$ 

VSH - problems (?)
Algebraic properties, e.g., easy to find messages with hash values h and 2h
Easy to invert hash for messages of small length
VSH has multiplicative property (Saarinen 2006): H(z)H(a∨b) = H(a)H(b) mod n, for z the all-zero bit string, a∧b = z, and |z| = |a| = |b|.
Someone must choose n such that p, q remain secret

VSH

Based on number-theoretic problems

Dakota

#### Introduction

Dakota

17 / 30

vsн

## VSH - defense

- Designers of VSH only aim for collision-resistance.
- VSH not to be used as replacement for random oracle nor SHA-1

"random oracles do not exist in the real world, and therefore relying on them too much is not recommended"

- Potential use in schemes which require only collision-resistance, example, Cramer-Shoup signatures, which relies on strong RSA assumption and collision-resistant hash function
- One of the best attempts to build hash function on number-theoretic problem so far..

## Based on factoring (Goldwasser, Micali, Rivest)

- N = pq,  $p \neq q$ , large primes,  $a_0, a_1$  random squares modulo N
- Public:  $N, a_0, a_1$

Introduction

- $h: \{0,1\} \times Z_N^* \to Z_N^*$
- $h(b, y) = a_b y^2 \mod N$
- Collision gives x, x' such that  $x^2 = x'^2 \mod N \rightarrow$  factoring

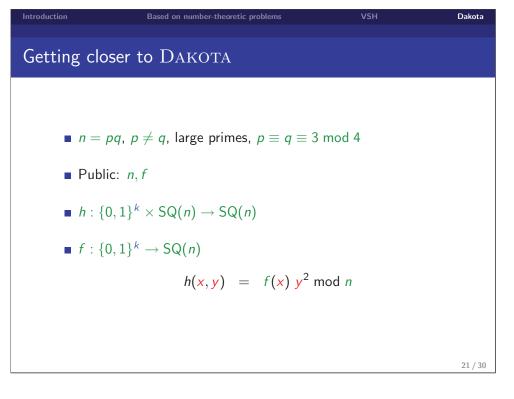
• More efficient variants with more squares  $a_0, \ldots, a_k$ , Damgård

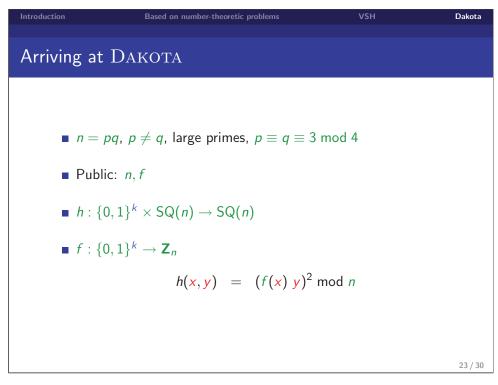
19/30

Dakota

20/30

Introduction	Based on number-theoretic problems	VSH	Dakota	Introduction	Based on number-theoretic problems
DaKoTa				Getting to	Дакота
• C • K • T	oTa, a hash function co-designed b Damgård, Ivan B. Knudsen, Lars R. Thomsen, Søren S. combination of modular arithmetic	-	/pto 18/30	<ul> <li>Publ</li> <li>h: {</li> </ul>	$pq, p \neq q$ , large primes, $p \equiv q \equiv 3 \mod 4$ lic: $n, f$ $0, 1\} \times SQ(n) \rightarrow SQ(n)$ $0, 1\} \rightarrow SQ(n)$ $h(b, y) = f(b) y^2 \mod n$





Introduction 251 Dakota Getting even closer to DAKOTA •  $n = pq, p \neq q$ , large primes,  $p \equiv q \equiv 3 \mod 4$ • Public: n, f•  $h: \{0,1\}^k \times SQ(n) \rightarrow SQ(n)$ •  $f: \{0,1\}^k \rightarrow Z_n$  $h(x, y) = f(x)^2 y^2 \mod n$ 

# 

Dakota

DAKOTA- an iterated hash function  $h(x, y) = (f(x)y)^2 \mod n$ 

### Assumption

Consider probabilistic polynomial time algorithm with input f, n, and output  $x, \tilde{x}, z$ . Probability is negligible that

$$x \neq \tilde{x}$$
 and  $f(x)/f(\tilde{x}) = \pm z^2 \mod r$ 

#### Theorem

Hash function H is collision intractable under Assumption

find collision with prob  $\epsilon \rightarrow$  break Assumption with prob  $\epsilon/2$ .

Dakota

DAKOTA- Assumption

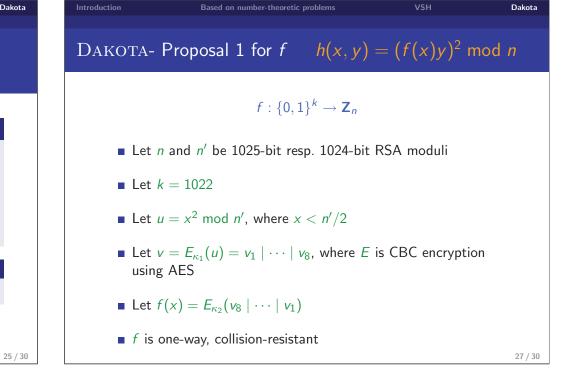
### Assumption

Consider probabilistic polynomial time algorithm with input f, n, and output  $x, \tilde{x}, z$ . Probability is negligible that

$$x \neq \tilde{x}$$
 and  $f(x)/f(\tilde{x}) = \pm z^2 \mod n$ 

 $h(x,y) = (f(x)y)^2 \mod n$ 

- f must be one-way: choose  $z, \tilde{x}$ , compute x
- f must be coll. resistant: find collision for f, let z = 1
- no circular argument?, since *f* does (need to) not compress



Introduction	Based on number-theoretic p	roblems	VSH	Dakota
Dakota- Pr	oposal 2 for <i>f</i>	h(x,y) =	$(f(x)y)^2$ mo	od <i>n</i>
	$f: \{0,1\}^k$	$\rightarrow \{0,1\}^k$		
■ Let <i>n</i> be	e 1025-bit RSA-modu	Ilus, let $k = 10$	)24	
• $f(x) = b$	$g(x)\oplus x$ , where $g$ is	permutation o	f 1024 bits	
■ proposa ■ trar	for g: form x into $8 \times 8$ mat	rix A with 16-bi	t values	
Do	4 times	$A \leftarrow E(A)^T,$		
wh	ere <i>E</i> is AES encryptic	on (fixed key) of	every column	

28/30

Introduction	Based on number-theoretic problems	VSH

# DAKOTA- Performance

Hash function	Approximate speed (cycles/byte)		
	32-bit	64-bit	
SHA-256	20	20	
VSH	840	?	
DAKOTA (Proposal 1)	385	170	
DAKOTA (Proposal 2)	330	170	

29 / 30

Dakota

VSH

Dakota

# Concluding remarks

- 1980s: Hash functions based on block ciphers
- 1990s:
  - Dedicated, faster hash functions (Rivest-kickoff)
  - Many broken block cipher based hash function proposals
- 2000s:
  - Many dedicated schemes have been broken in later years
  - Many new constructions
- Renaissance of block cipher based proposals
- Renaissance of constructions with proofs of security
- SHA-3, likely to become "big SHA-1", speed issue