Everybody's a Target: Scalability in Public-Key Encryption

Benedikt Auerbach[1] Federico Giacon[2] Eike Kiltz[3]

[1]IST Austria, Klosterneuburg, Austria
[2]Gnosis Service GmbH, Berlin, Germany
[3]Horst Görtz Institut für IT-Sicherheit, Ruhr-Universität Bochum, Germany

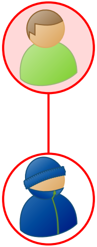
May 04, 2020

Agenda

- multi-instance security and the scaling factor
- the scaling behavior of Hashed-ElGamal key encapsulation
- generic group lower bounds for multi-instance CDH-type problems

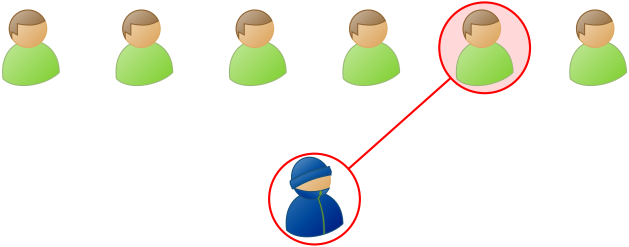
Multi-instance security

- usual security definition for cryptographic schemes
  - adversary unable to compromise a *single* user



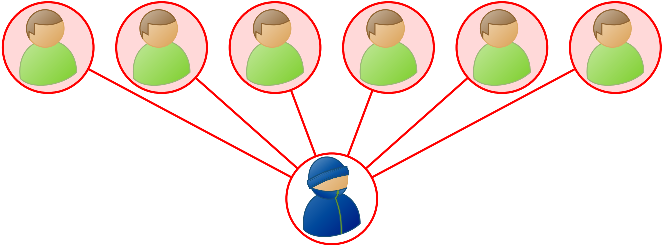
Multi-instance security

- usual security definition for cryptographic schemes
  - adversary unable to compromise a *single* user

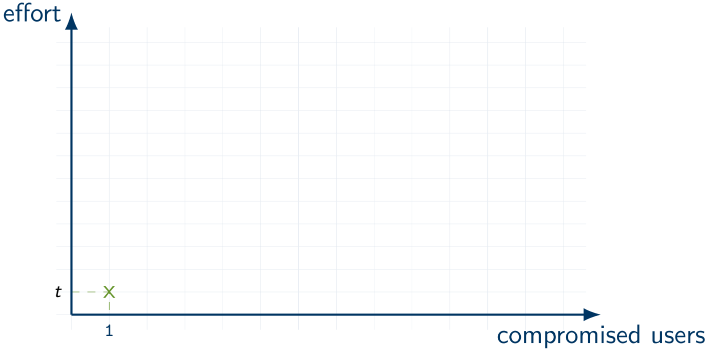


Multi-instance security

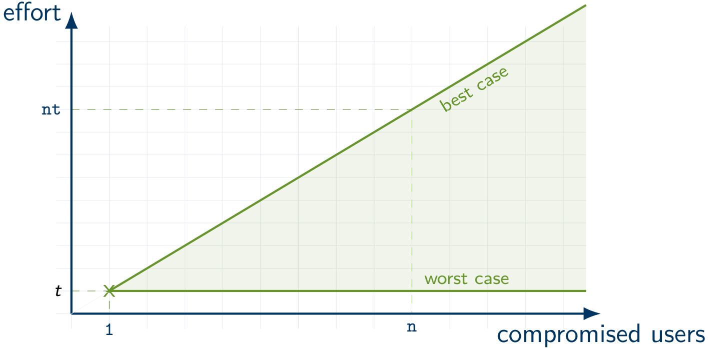
- usual security definition for cryptographic schemes
  - adversary unable to compromise a *single* user
- this work: scaling of security in the number of users
  - how much more computational effort does it take to compromise *all* of n users compared to compromising one?



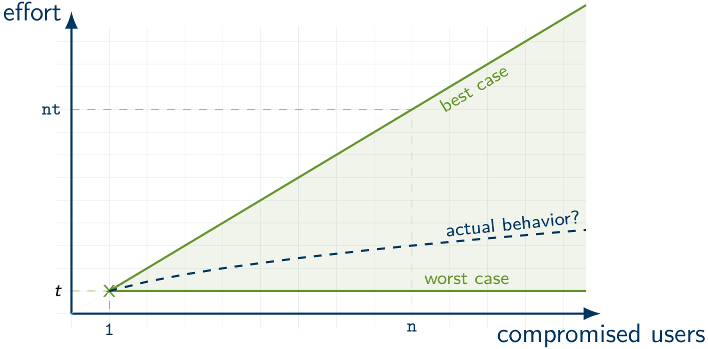
Scaling behavior of cryptographic schemes



Scaling behavior of cryptographic schemes



Scaling behavior of cryptographic schemes



Background

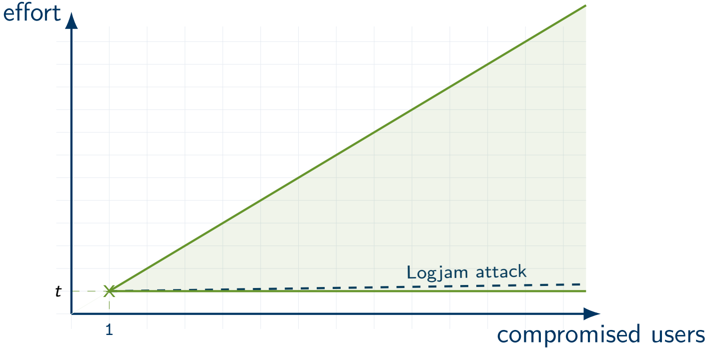
- theory: parameters of schemes chosen such that even breaking a *single* instance is infeasible
  - in particular impossible to break many instances
- practice: use of outdated parameters widespread
  - breaking of single instance within reach
  - bad scaling behavior could enable large-scale attack

Logjam attack

- bad scaling-behavior exploited in Logjam attack [ADGG+15]
  - attacked TLS in the finite-field setting for primes of length 512
  - effort to break 2^{20} instances only doubles compared to breaking one

Logjam attack

Scaling behavior of ElGamal for subgroups of $\mathbb{F}_p^*$, p prime of length 512



Effort to break 2^{20} instances only doubles compared to breaking one

Our contributions

- scaling behavior; theoretical perspective
  - adapt multi-instance security to key-encapsulation mechanisms
  - define the scaling factor of schemes
- scaling behavior; application to Hashed-ElGamal (HEG) key encapsulation
  - consider HEG for different parameter settings
  - compute scaling factor in idealized models

Multi-Instance Security and the Scaling Factor

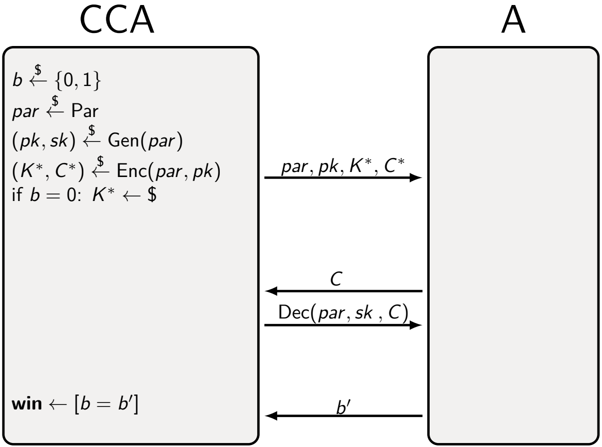
Reminder: key-encapsulation mechanisms

- Key-encapsulation mechanism KEM consists of algorithms

$$
\begin{aligned}
par &\xleftarrow{\$} \mathsf{Par} \\
(pk, sk) &\xleftarrow{\$} \mathsf{Gen}(par) \\
(K, C) &\xleftarrow{\$} \mathsf{Enc}(par, pk) \\
K &\leftarrow \mathsf{Dec}(par, sk, C)
\end{aligned}
$$

Security notions for KEMs

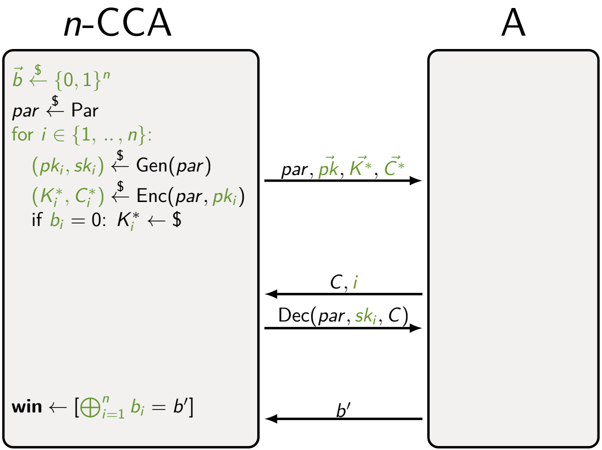
CCA: single-instance setting



Advantage: $\mathrm{Adv}^{\mathsf{CCA}}_{\mathsf{KEM}}(\mathsf{A}) = \Pr[\mathbf{win}] - 1/2$

Security notions for KEMs

n-CCA: multi-instance setting [BelRisTes12]



Advantage: $\mathrm{Adv}_{\mathsf{KEM}}^{n\text{-CCA}}(\mathsf{A}) = \Pr[\textbf{win}] - 1/2$

Scaling factor

- how does the security of a key-encapsulation mechanism (KEM) scale in the number of users?
  - we define the scaling factor of KEM

$$\mathrm{SF}(n) = \frac{\mathrm{MinTime}(n)}{\mathrm{MinTime}(1)}$$

  - $\mathrm{MinTime}(n)$: running time of fastest adversary breaking n-CCA security users with success probability 1

Scaling factor

- how does the security of a key-encapsulation mechanism (KEM) scale in the number of users?
  - we define the scaling factor of KEM

$$\mathrm{SF}(n) = \frac{\mathrm{MinTime}(n)}{\mathrm{MinTime}(1)}$$

  - $\mathrm{MinTime}(n)$: running time of fastest adversary breaking n-CCA security users with success probability 1

Lemma

$$1 \leq \mathrm{SF}(n) \leq n$$

The Scaling Behavior of Hashed-ElGamal

Overview on our results

- considered KEM: Hashed-ElGamal
  - consider variants with different shared parameters (granularity)
  - elliptic-curve setting
  - bounds in generic-group model and random-oracle model
- $\mathbb{G}$ group of prime order p generated by g

Granularity	*par*	*sk*	*pk*	$\mathrm{SF}_{\mathsf{HEG}}(n)$
high	$(\mathbb{G}, p, g)$	x	g^x	$\Theta(\sqrt{n})$
medium	$(\mathbb{G}, p)$	(g, x)	(g, g^x)	$\Theta(\sqrt{n})$
low	$\perp$	$((\mathbb{G}, p, g), x)$	$((\mathbb{G}, p, g), g^x)$	$\Theta(n)$

Overview on our results

- goal: bound $\mathrm{SF}_{\mathsf{HEG}}(n) = \frac{\mathrm{MinTime}(n)}{\mathrm{MinTime}(1)}$

Overview on our results

- goal: bound $\mathrm{SF}_{\mathsf{HEG}}(n) = \frac{\mathrm{MinTime}(n)}{\mathrm{MinTime}(1)}$
- upper bound
  - known generic algorithms:
    $$\mathrm{MinTime}(n) = \begin{cases} O(\sqrt{np}) & \text{high/med. granularity} \\ O(n\sqrt{p}) & \text{low granularity} \end{cases}$$
  - known generic bound: $\mathrm{MinTime}(1) = \Omega(\sqrt{p})$

Overview on our results

- goal: bound $\mathrm{SF}_{\mathsf{HEG}}(n) = \frac{\mathrm{MinTime}(n)}{\mathrm{MinTime}(1)}$
- upper bound
  - known generic algorithms:
    $$\mathrm{MinTime}(n) = \begin{cases} O(\sqrt{np}) & \text{high/med. granularity} \\ O(n\sqrt{p}) & \text{low granularity} \end{cases}$$
  - known generic bound: $\mathrm{MinTime}(1) = \Omega(\sqrt{p})$
- lower bound
  - known generic algorithm: $\mathrm{MinTime}(1) = O(\sqrt{p})$
  - this work: generic-group bounds
    $$\mathrm{MinTime}(n) = \begin{cases} \Omega(\sqrt{np}) & \text{high/med. granularity} \\ \Omega(n\sqrt{p}) & \text{low granularity} \end{cases}$$

Generic-group lower bound on $\mathrm{MinTime}_{\mathsf{HEG}}(n)$

Overview

n-$\mathsf{CCA}_{\mathsf{HEG}}$

Generic-group lower bound on $\mathrm{MinTime}_{\mathsf{HEG}}(n)$

Overview

$$n\text{-gapCDH} \overset{\text{ROM}}{\Longrightarrow} n\text{-CCA}_{\mathsf{HEG}}$$

ROM $\sim$ random-oracle model
n-gapCDH $\sim$ multi-instance gap Diffie-Hellman problem

Generic-group lower bound on $\mathrm{MinTime}_{\mathsf{HEG}}(n)$

Overview

$$n\text{-gapDL} \xRightarrow{\text{AGM}} n\text{-gapCDH} \xRightarrow{\text{ROM}} n\text{-CCA}_{\mathsf{HEG}}$$

ROM $\sim$ random-oracle model

n-gapCDH $\sim$ multi-instance gap Diffie-Hellman problem

AGM $\sim$ algebraic-group model [FKL18]

n-gapDL $\sim$ multi-instance gap discrete-logarithm problem

Generic-group lower bound on $\mathrm{MinTime}_{\mathsf{HEG}}(n)$

Overview

$$\xrightarrow{\text{GGM}} \quad n\text{-gapDL} \quad \xrightarrow{\text{(AGM)}} \quad n\text{-gapCDH} \quad \xrightarrow{\text{ROM}} \quad n\text{-CCA}_{\mathsf{HEG}}$$

ROM $\sim$ random-oracle model

n-gapCDH $\sim$ multi-instance gap Diffie-Hellman problem

AGM $\sim$ algebraic-group model [FKL18]

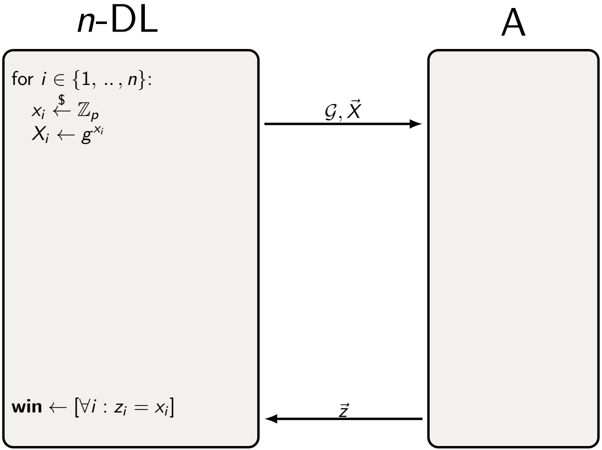
n-gapDL $\sim$ multi-instance gap discrete-logarithm problem

GGM $\sim$ generic-group model

Generic-Group Lower Bounds for Multi-Instance CDH-Type Problems

Multi-instance CDH-type problems

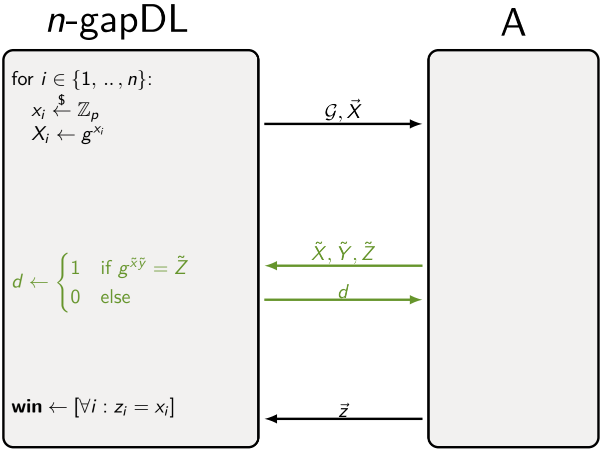
Multi-instance discrete logarithm problem, $\mathcal{G} = (\mathbb{G}, p, g)$



Advantage: $\mathrm{Adv}^{n\text{-DL}}(\mathsf{A}) = \Pr[\mathbf{win}]$

Multi-instance CDH-type problems

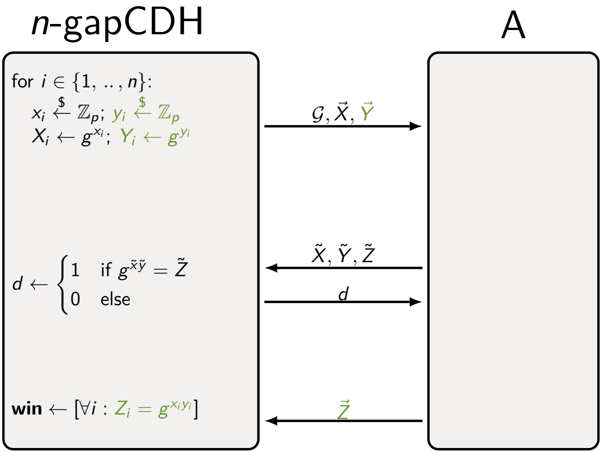
Multi-instance *gap* discrete logarithm problem, $\mathcal{G} = (\mathbb{G}, p, g)$



Advantage: $\mathrm{Adv}^{n\text{-gapDL}}(\mathsf{A}) = \Pr[\textbf{win}]$

Multi-instance CDH-type problems

Multi-instance gap computational Diffie-Hellman problem, $\mathcal{G} = (\mathbb{G}, p, g)$



Advantage: $\mathrm{Adv}^{n\text{-gapCDH}}(\mathsf{A}) = \Pr[\mathbf{win}]$

Multi-instance generic-group lower bounds

Overview

problem	granularity	MinTime	
n-DL	high	$\Omega(\sqrt{np})$	[Yun15]
n-DL	low	$\Omega(\sqrt{np})$	[GDJY13]

Generic-group bounds for multi-instance Diffie-Hellman-type problems

- $\mathbb{G}$ of prime order p
- n instances

Multi-instance generic-group lower bounds

Overview

problem	granularity	MinTime	
n-DL	high	$\Omega(\sqrt{np})$	[Yun15]
n-DL	low	$\Omega(\sqrt{np})$	[GDJY13]
			this work
n-gapDL	high/med.	$\Omega(\sqrt{np})$	
n-gapCDH	high/med.	$\Omega(\sqrt{np})$	
n-gapDL	low	$\Omega(n\sqrt{p})$	
n-gapCDH	low	$\Omega(n\sqrt{p})$	

Generic-group bounds for multi-instance Diffie-Hellman-type problems

- $\mathbb{G}$ of prime order p
- n instances

Multi-instance generic-group lower bounds

Overview

problem	granularity	MinTime	
n-DL	high	$\Omega(\sqrt{np})$	[Yun15]
n-DL	low	$\Omega(\sqrt{np})$	[GDJY13]
			this work
n-gapDL	high/med.	$\Omega(\sqrt{np})$	
n-gapCDH	high/med.	$\Omega(\sqrt{np})$	
n-gapDL	low	$\Omega(n\sqrt{p})$	
n-gapCDH	low	$\Omega(n\sqrt{p})$	
n-polyDL$_d$	high	$\Omega(\sqrt{np/d})$	

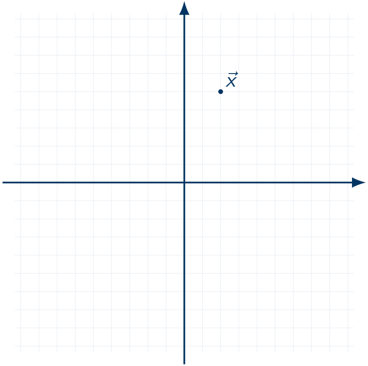
Generic-group bounds for multi-instance Diffie-Hellman-type problems

- $\mathbb{G}$ of prime order p
- n instances

Intuition behind proofs

n-gapDL

- high granularity
  - reduce *n*-gapDL to geometric search problem: search-by-hypersurface problem (SHS_2)
  - prove information theoretic bound on hardness of SHS_2
  - DDH-oracle requires us to work in realm of commutative algebra

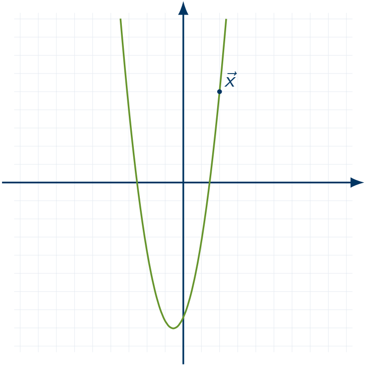


space: $\mathbb{Z}_p^n$; goal: find $\vec{x}$

Intuition behind proofs

n-gapDL

- high granularity
  - reduce n-gapDL to geometric search problem: search-by-hypersurface problem (SHS_2)
  - prove information theoretic bound on hardness of SHS_2
  - DDH-oracle requires us to work in realm of commutative algebra

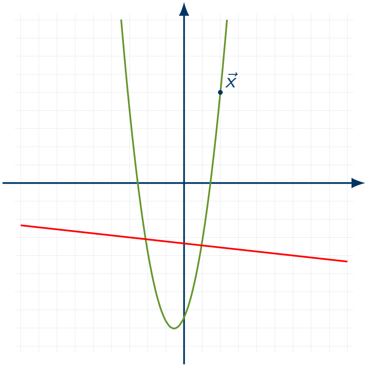


space: $\mathbb{Z}_p^n$; goal: find $\vec{x}$

Intuition behind proofs

n-gapDL

- high granularity
  - reduce n-gapDL to geometric search problem: search-by-hypersurface problem (SHS_2)
  - prove information theoretic bound on hardness of SHS_2
  - DDH-oracle requires us to work in realm of commutative algebra

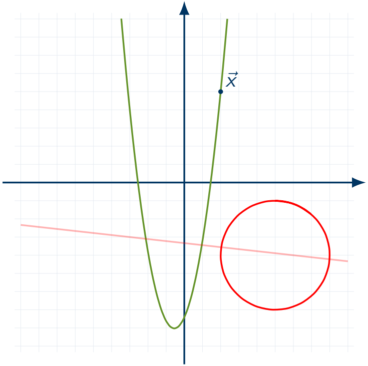


space: $\mathbb{Z}_p^n$; goal: find $\vec{x}$

Intuition behind proofs

n-gapDL

- high granularity
  - reduce n-gapDL to geometric search problem: search-by-hypersurface problem (SHS_2)
  - prove information theoretic bound on hardness of SHS_2
  - DDH-oracle requires us to work in realm of commutative algebra

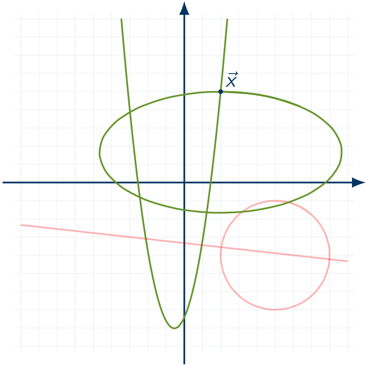


space: $\mathbb{Z}_p^n$; goal: find $\vec{x}$

Intuition behind proofs

n-gapDL

- high granularity
  - reduce n-gapDL to geometric search problem: search-by-hypersurface problem (SHS_2)
  - prove information theoretic bound on hardness of SHS_2
  - DDH-oracle requires us to work in realm of commutative algebra

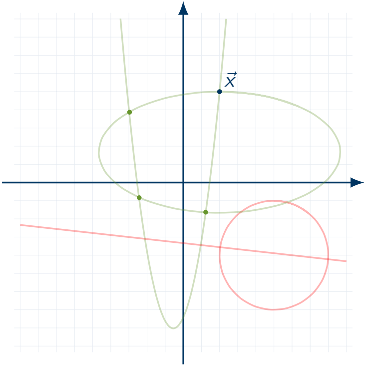


space: $\mathbb{Z}_p^n$; goal: find $\vec{x}$

Intuition behind proofs

n-gapDL

- high granularity
  - reduce n-gapDL to geometric search problem: search-by-hypersurface problem (SHS_2)
  - prove information theoretic bound on hardness of SHS_2
  - DDH-oracle requires us to work in realm of commutative algebra

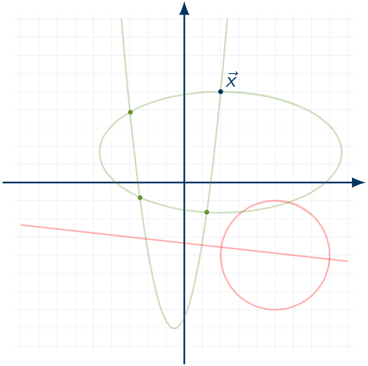


space: $\mathbb{Z}_p^n$; goal: find $\vec{x}$

Intuition behind proofs

n-gapDL

- high granularity
  - reduce *n*-gapDL to geometric search problem: search-by-hypersurface problem (SHS_2)
  - prove information theoretic bound on hardness of SHS_2
  - DDH-oracle requires us to work in realm of commutative algebra
- low / medium granularity
  - derived from high granularity result



space: $\mathbb{Z}_p^n$; goal: find $\vec{x}$

Intuition behind proofs

n-gapCDH

- high granularity
  - show that bound for n-gapDL carries over to n-gapCDH using AGM

Intuition behind proofs

n-gapCDH

- high granularity
  - show that bound for *n*-gapDL carries over to *n*-gapCDH using AGM
- low / medium granularity
  - derived from high granularity result

Summary and Future Directions

- summary
  - we define the *scaling factor* SF, which measures the scaling of a scheme's security in the number of users
  - we compute lower bounds on SF for variants of the Hashed-ElGamal KEM in the generic-group model
  - we prove generic lower bounds on the hardness of various multi-instance CDH-type problems
- future directions
  - revisit the KEM-DEM paradigm
  - consider preprocessing

ia.cr/2019/364

References I

David Adrian, Karthikeyan Bhargavan, Zakir Durumeric, Pierrick Gaudry, Matthew Green, J. Alex Halderman, Nadia Heninger, Drew Springall, Emmanuel Thomé, Luke Valenta, Benjamin VanderSloot, Eric Wustrow, Santiago Zanella Béguelin, and Paul Zimmermann.
Imperfect forward secrecy: How Diffie-Hellman fails in practice.
In Indrajit Ray, Ninghui Li, and Christopher Kruegel:, editors, *ACM CCS 15*, pages 5–17. ACM Press, October 2015.

Mihir Bellare, Thomas Ristenpart, and Stefano Tessaro.
Multi-instance security and its application to password-based cryptography.
In Reihaneh Safavi-Naini and Ran Canetti, editors, *CRYPTO 2012*, volume 7417 of *LNCS*, pages 312–329. Springer, Heidelberg, August 2012.

References II

Georg Fuchsbauer, Eike Kiltz, and Julian Loss.
The algebraic group model and its applications.
In Hovav Shacham and Alexandra Boldyreva, editors, *CRYPTO 2018, Part II*, volume 10992 of *LNCS*, pages 33–62. Springer, Heidelberg, August 2018.

Juan A. Garay, David S. Johnson, Aggelos Kiayias, and Moti Yung.
Resource-based corruptions and the combinatorics of hidden diversity.
In Robert D. Kleinberg, editor, *ITCS 2013*, pages 415–428. ACM, January 2013.

Aaram Yun.
Generic hardness of the multiple discrete logarithm problem.
In Elisabeth Oswald and Marc Fischlin, editors, *EUROCRYPT 2015, Part II*, volume 9057 of *LNCS*, pages 817–836. Springer, Heidelberg, April 2015.