Password-Authenticated Key Exchange from Group Actions

Michel Abdalla[1,2], Thorsten Eisenhofer[3], Eike Kiltz[3], Sabrina Kunzweiler[3], Doreen Riepel[3]

August 15, 2022

[1]DIENS, École normale supérieure, CNRS, PSL University, Paris, France
[2]DFINITY, Zürich, Switzerland
[3]Ruhr-Universität Bochum, Germany

Motivation

Password Authenticated Key Exchange

- Establish a session key based on
  a (potentially weak) password

Motivation

Password Authenticated Key Exchange

- Establish a session key based on a (potentially weak) password

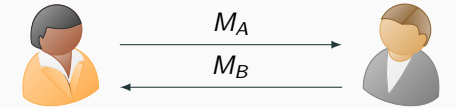
Alice (pw)

Bob (pw)

Motivation

Password Authenticated Key Exchange

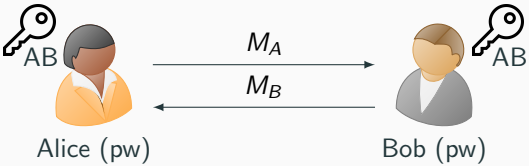
- Establish a session key based on a (potentially weak) password



Motivation

Password Authenticated Key Exchange

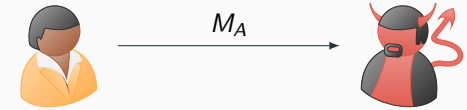
- Establish a session key based on a (potentially weak) password



Motivation

Password Authenticated Key Exchange

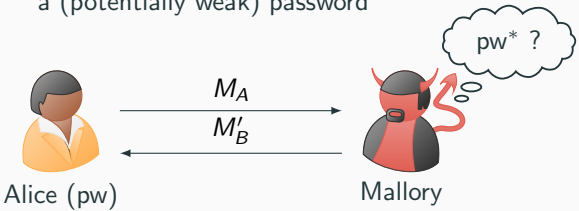
- Establish a session key based on a (potentially weak) password



Motivation

Password Authenticated Key Exchange

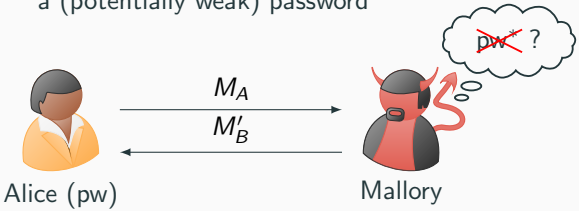
- Establish a session key based on a (potentially weak) password



Motivation

Password Authenticated Key Exchange

- Establish a session key based on a (potentially weak) password

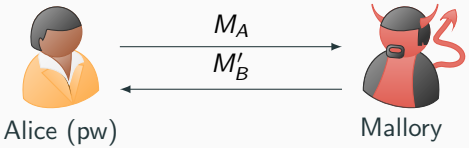


- Best attack: online dictionary attack

Motivation

Password Authenticated Key Exchange

- Establish a session key based on a (potentially weak) password



- Best attack: online dictionary attack

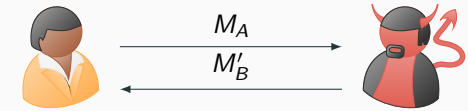
Group Actions

- Abstraction is close to the classical DH-setting

Motivation

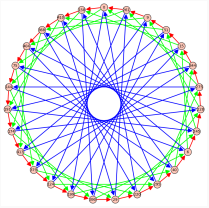
Password Authenticated Key Exchange

- Establish a session key based on a (potentially weak) password



- Best attack: online dictionary attack

Group Actions

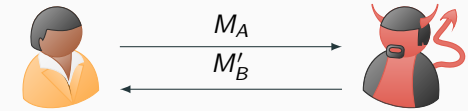


- Abstraction is close to the classical DH-setting
- CSIDH as candidate for post-quantum security
- Public-Key Encryption, Signatures, Oblivious Transfer, ...

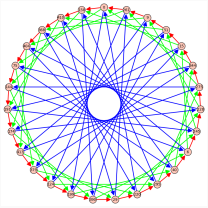
Motivation

Password Authenticated Key Exchange from **Group Actions** ?

- Establish a session key based on a (potentially weak) password



- Best attack: online dictionary attack



- Abstraction is close to the classical DH-setting
- CSIDH as candidate for post-quantum security
- Public-Key Encryption, Signatures, Oblivious Transfer, ...

Motivation

Password Authenticated Key Exchange from **Group Actions** ?

Motivation

Password Authenticated Key Exchange from **Group Actions** ?

Difficulties (e.g., [AJK$^+$20])

- Limited structure of the group action
- Special properties of CSIDH

$\Rightarrow$ Known DH-based constructions cannot be directly translated to the group action setting

Motivation

Password Authenticated Key Exchange from **Group Actions** ?

Difficulties (e.g., [AJK$^+$20])

- Limited structure of the group action
- Special properties of CSIDH

$\Rightarrow$ Known DH-based constructions cannot be directly translated to the group action setting

Generic Constructions

- Quite inefficient construction using OT
- Unclear how to use the HPS of [ADMP20]

Cryptographic Group Actions

Cryptographic Group Actions [ADMP20]

Group Action

Let $(\mathcal{G}, \cdot)$ be a group with identity element $id \in \mathcal{G}$, and $\mathcal{X}$ a set. A map $\star : \mathcal{G} \times \mathcal{X} \to \mathcal{X}$ is a group action if it satisfies the following properties:

1. Identity: $id \star x = x$ for all $x \in \mathcal{X}$.
2. Compatibility: $(g \cdot h) \star x = g \star (h \star x)$ for all $g, h \in \mathcal{G}$ and $x \in \mathcal{X}$.

Cryptographic Group Actions [ADMP20]

Group Action

Let $(\mathcal{G}, \cdot)$ be a group with identity element $id \in \mathcal{G}$, and $\mathcal{X}$ a set. A map $\star : \mathcal{G} \times \mathcal{X} \to \mathcal{X}$ is a group action if it satisfies the following properties:

1. Identity: $id \star x = x$ for all $x \in \mathcal{X}$.
2. Compatibility: $(g \cdot h) \star x = g \star (h \star x)$ for all $g, h \in \mathcal{G}$ and $x \in \mathcal{X}$.

Technical Assumptions

- $\mathcal{G}$ and $\mathcal{X}$ are finite.
- $\mathcal{G}$ is commutative.
- $\star : \mathcal{G} \times \mathcal{X} \to \mathcal{X}$ is regular.
- A distinguished element $\tilde{x} \in \mathcal{X}$ ("origin").

Cryptographic Group Actions [ADMP20]

Group Action

Let $(\mathcal{G}, \cdot)$ be a group with identity element $id \in \mathcal{G}$, and $\mathcal{X}$ a set. A map $\star : \mathcal{G} \times \mathcal{X} \to \mathcal{X}$ is a group action if it satisfies the following properties:

1. Identity: $id \star x = x$ for all $x \in \mathcal{X}$.
2. Compatibility: $(g \cdot h) \star x = g \star (h \star x)$ for all $g, h \in \mathcal{G}$ and $x \in \mathcal{X}$.

Technical Assumptions

- $\mathcal{G}$ and $\mathcal{X}$ are finite.
- $\mathcal{G}$ is commutative.
- $\star : \mathcal{G} \times \mathcal{X} \to \mathcal{X}$ is regular.
- A distinguished element $\tilde{x} \in \mathcal{X}$ ("origin").

⚠ In general, we cannot combine two elements of the set $\mathcal{X}$!

The CSIDH Group Action

CSIDH [CLM+18] can be seen as a *restricted effective* group action:

$\mathcal{G} =$ corresponds to isogenies between elliptic curves

$\mathcal{X} =$ supersingular elliptic curves over $\mathbb{F}_p$

The CSIDH Group Action

CSIDH [CLM+18] can be seen as a *restricted effective* group action:

$\mathcal{G} =$ corresponds to isogenies between elliptic curves

$\mathcal{X} =$ supersingular elliptic curves over $\mathbb{F}_p$

Computational Assumptions

- DLOG: Given $g \star \tilde{x} \in \mathcal{X}$, it is hard to find $g \in \mathcal{G}$.
- CDH: Given $(g \star \tilde{x}, h \star \tilde{x}) \in \mathcal{X}^2$, it is hard to find $z = gh \star \tilde{x} \in \mathcal{X}$.
- DDH: Given $(g \star \tilde{x}, h \star \tilde{x}, gh \star \tilde{x}) \in \mathcal{X}^3$ or $(g \star \tilde{x}, h \star \tilde{x}, u \star \tilde{x}) \in \mathcal{X}^3$, decide which is the case.

The CSIDH Group Action

CSIDH [CLM+18] can be seen as a *restricted effective* group action:

$\mathcal{G}$ = corresponds to isogenies between elliptic curves

$\mathcal{X}$ = supersingular elliptic curves over $\mathbb{F}_p$

Computational Assumptions

- DLOG: Given $g \star \tilde{x} \in \mathcal{X}$, it is hard to find $g \in \mathcal{G}$.
- CDH: Given $(g \star \tilde{x}, h \star \tilde{x}) \in \mathcal{X}^2$, it is hard to find $z = gh \star \tilde{x} \in \mathcal{X}$.
- DDH: Given $(g \star \tilde{x}, h \star \tilde{x}, gh \star \tilde{x}) \in \mathcal{X}^3$ or $(g \star \tilde{x}, h \star \tilde{x}, u \star \tilde{x}) \in \mathcal{X}^3$, decide which is the case.
- Strong/Gap CDH: same as CDH but with access to a decision oracle DDH, where

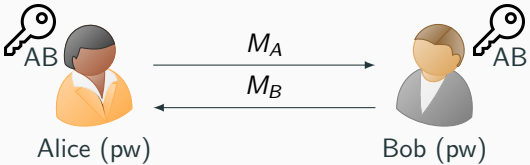
$$\mathrm{DDH}(x, y, z) = \begin{cases} 1 & \mathrm{CDH}(x, y) = z \\ 0 & \text{otherwise} \end{cases}$$

Password-Authenticated Key Exchange

Password-Authenticated Key Exchange (PAKE)

Focus

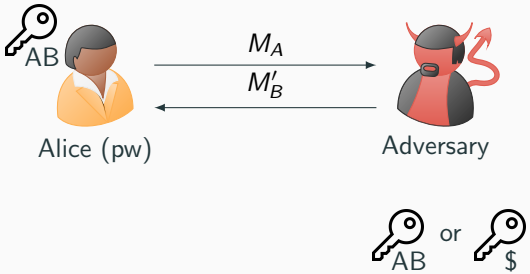
- balanced PAKE



Password-Authenticated Key Exchange (PAKE)

Focus

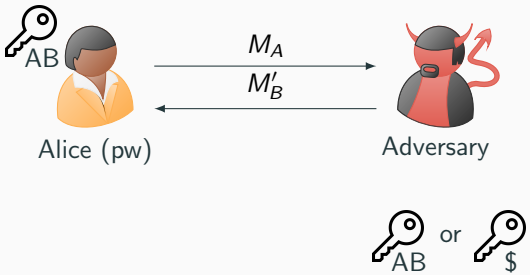
- balanced PAKE
- BPR security model (game-based) with extension to multiple test queries



Password-Authenticated Key Exchange (PAKE)

Focus

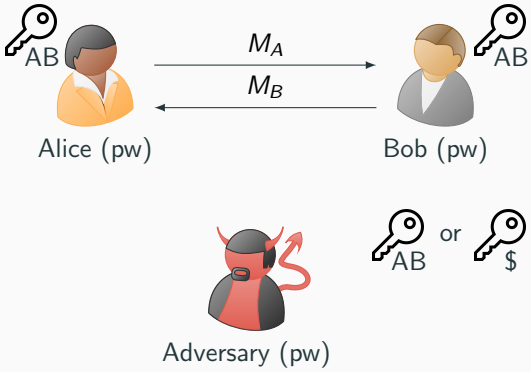
- balanced PAKE
- BPR security model (game-based) with extension to multiple test queries
- perfect forward secrecy



Password-Authenticated Key Exchange (PAKE)

Focus

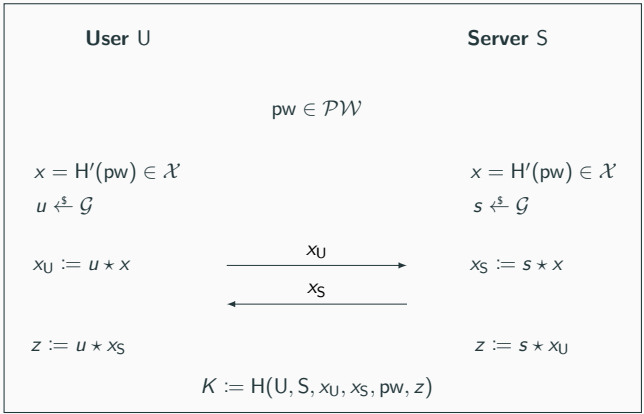
- balanced PAKE
- BPR security model (game-based) with extension to multiple test queries
- ~~perfect~~ weak forward secrecy



Our First Protocol

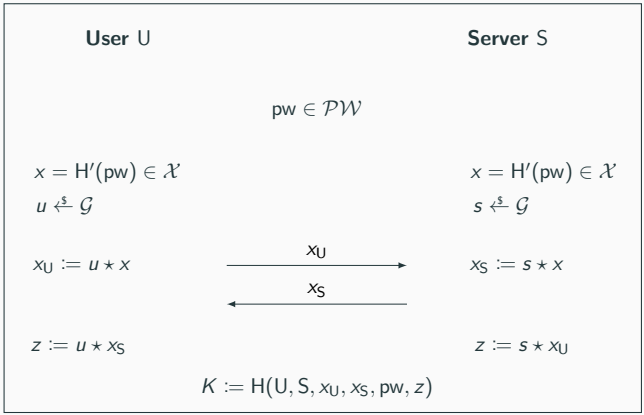
Starting Point: SPEKE

Simple Password Exponential Key Exchange [Jab96]



Starting Point: SPEKE

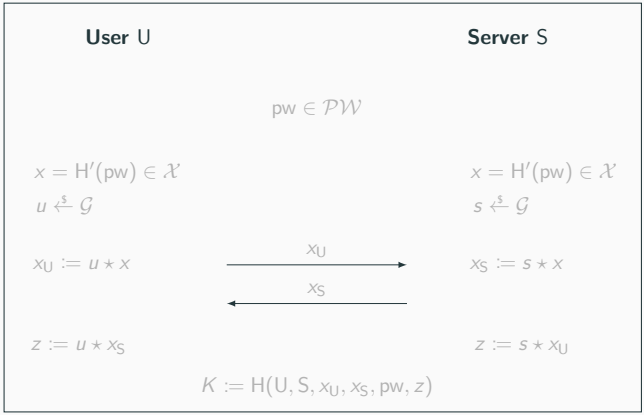
Simple Password Exponential Key Exchange [Jab96]



Problem: A hash function $\mathsf{H}' : \mathcal{PW} \to \mathcal{X}$ is still an open problem for CSIDH [BBD+22, MMP22].

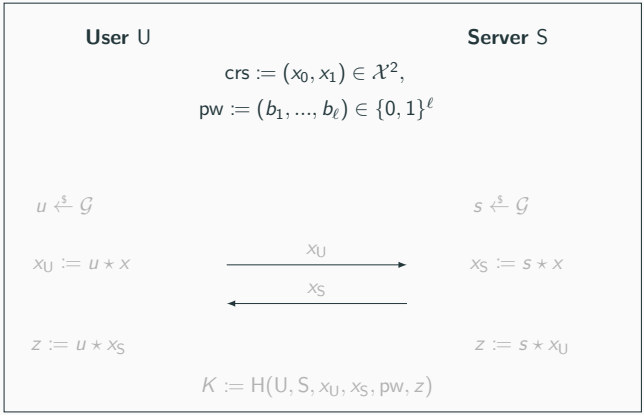
Our Group Action PAKE

Idea: Replace the hash function by a bit-by-bit approach.



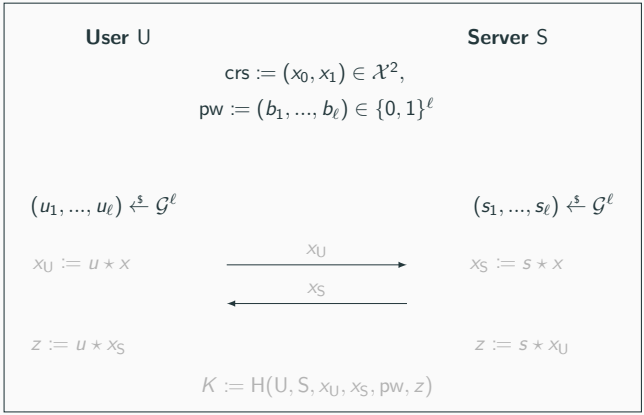
Our Group Action PAKE

Idea: Replace the hash function by a bit-by-bit approach.



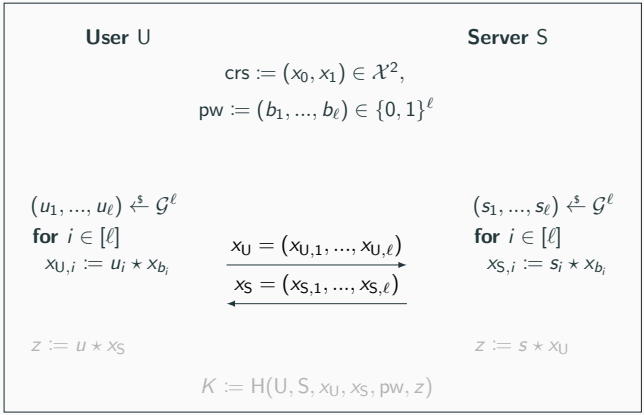
Our Group Action PAKE

Idea: Replace the hash function by a bit-by-bit approach.



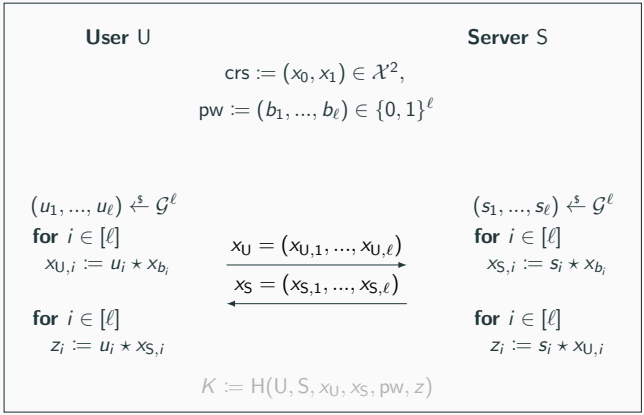
Our Group Action PAKE

Idea: Replace the hash function by a bit-by-bit approach.



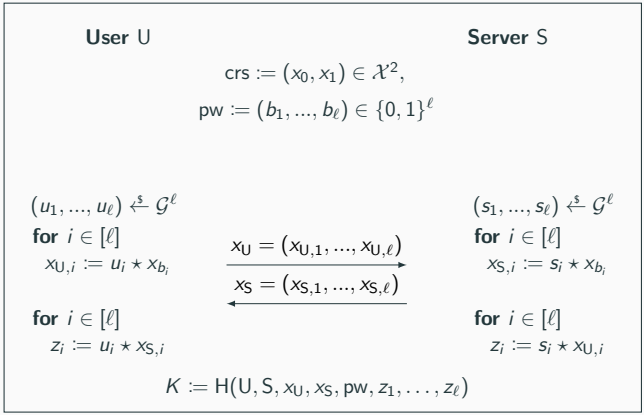
Our Group Action PAKE

Idea: Replace the hash function by a bit-by-bit approach.



Our Group Action PAKE

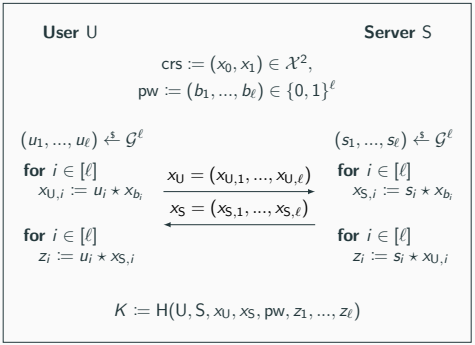
Idea: Replace the hash function by a bit-by-bit approach.



(In)Security of our Protocol

Security against Passive Adversaries

- secure under Strong CDH + ROM



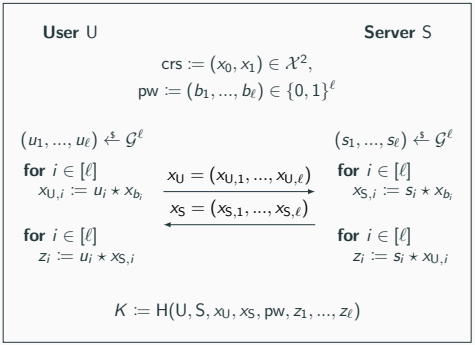
(In)Security of our Protocol

Security against Passive Adversaries

- secure under Strong CDH + ROM

Security against Active Adversaries

- secure under (Strong) Simultaneous DH + ROM
- but: insecure when instantiated with CSIDH



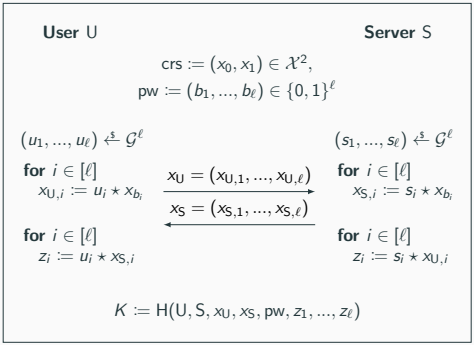
(In)Security of our Protocol

Security against Passive Adversaries

- secure under Strong CDH + ROM

Security against Active Adversaries

- secure under (Strong) Simultaneous DH + ROM
- but: insecure when instantiated with CSIDH

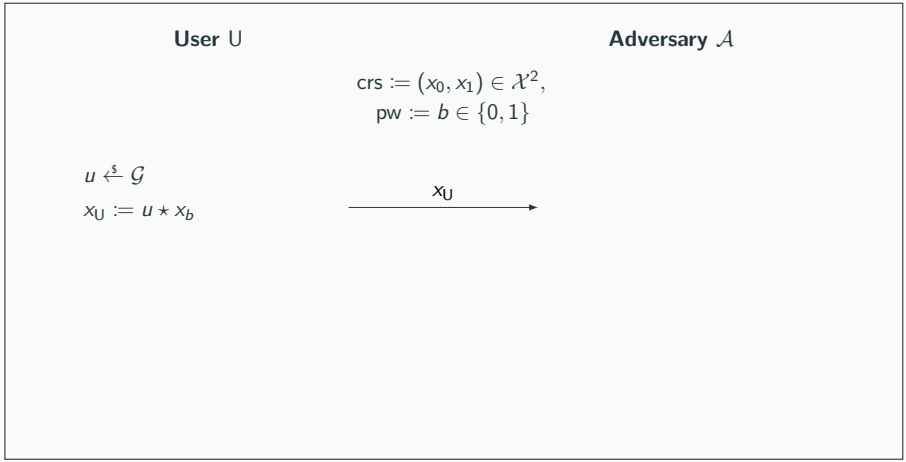


Additional Structure of the CSIDH Group Action

- For any $x \in \mathcal{X}$, we can efficiently compute its *twist* denoted by x^t.
- Let $x = g \star \tilde{x}$, then $x^t = g^{-1} \star \tilde{x}$. In particular $\tilde{x}^t = \tilde{x}$.

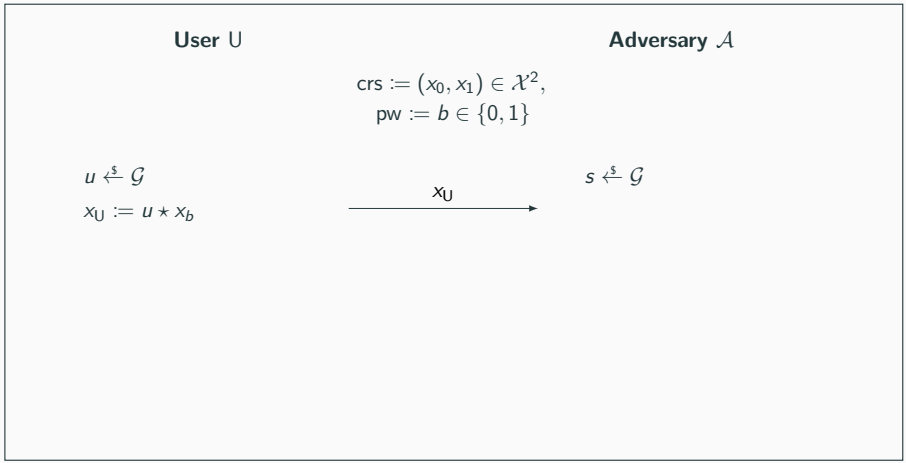
(In)Security of our Protocol

Twists yield an Offline Dictionary Attack!



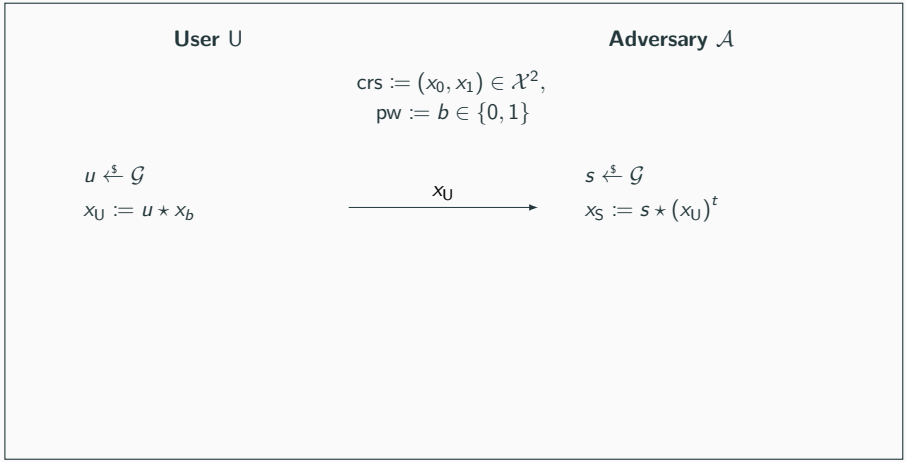
(In)Security of our Protocol

Twists yield an Offline Dictionary Attack!



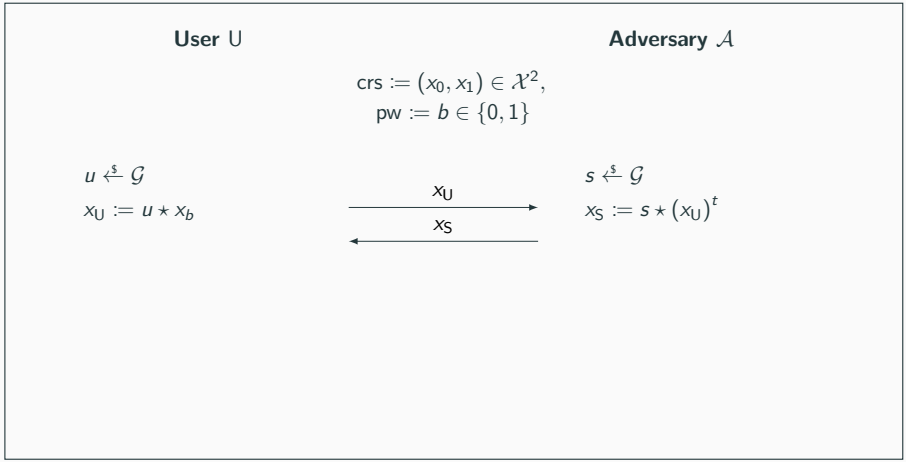
(In)Security of our Protocol

Twists yield an Offline Dictionary Attack!



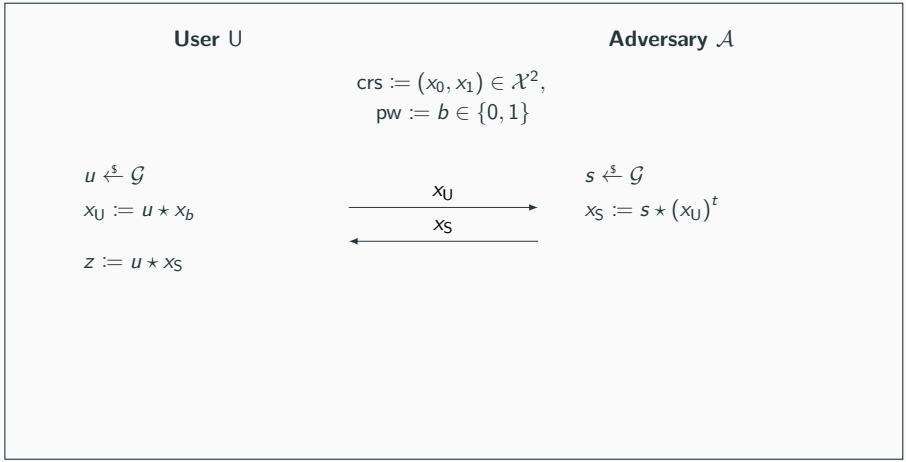
(In)Security of our Protocol

Twists yield an Offline Dictionary Attack!



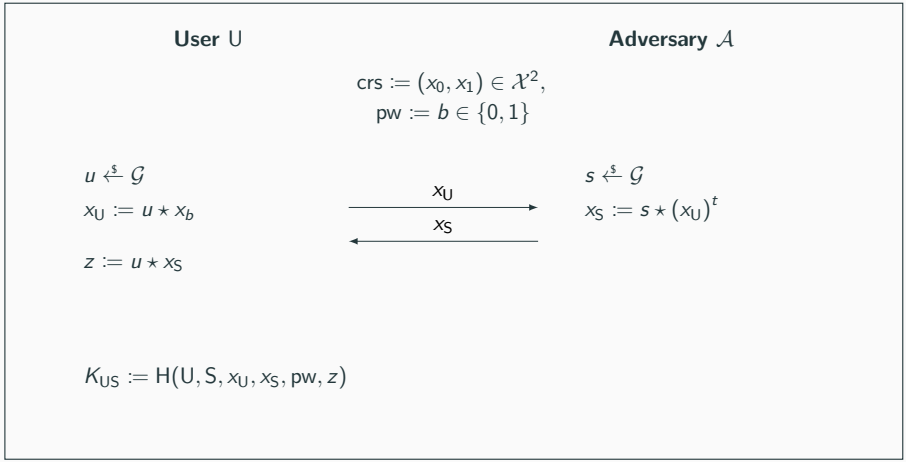
(In)Security of our Protocol

Twists yield an Offline Dictionary Attack!



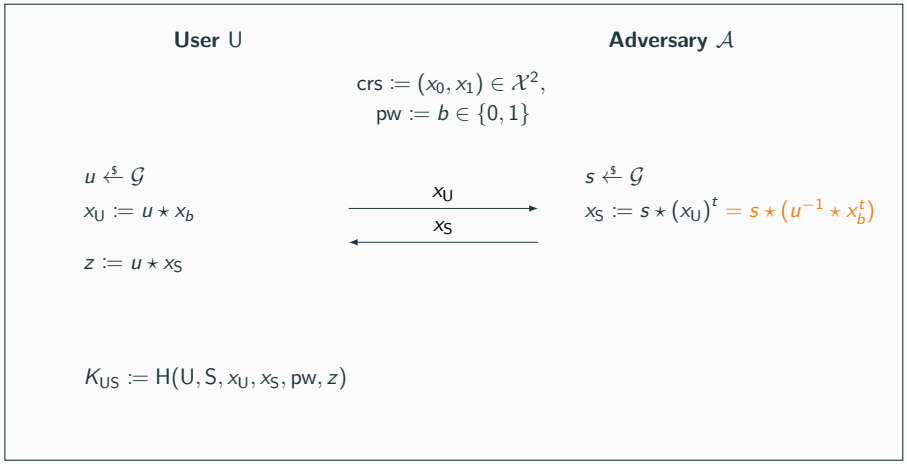
(In)Security of our Protocol

Twists yield an Offline Dictionary Attack!



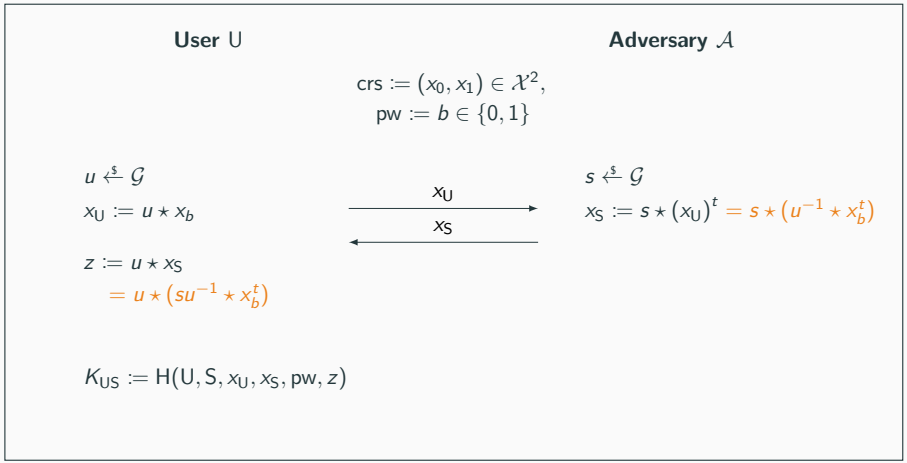
(In)Security of our Protocol

Twists yield an Offline Dictionary Attack!



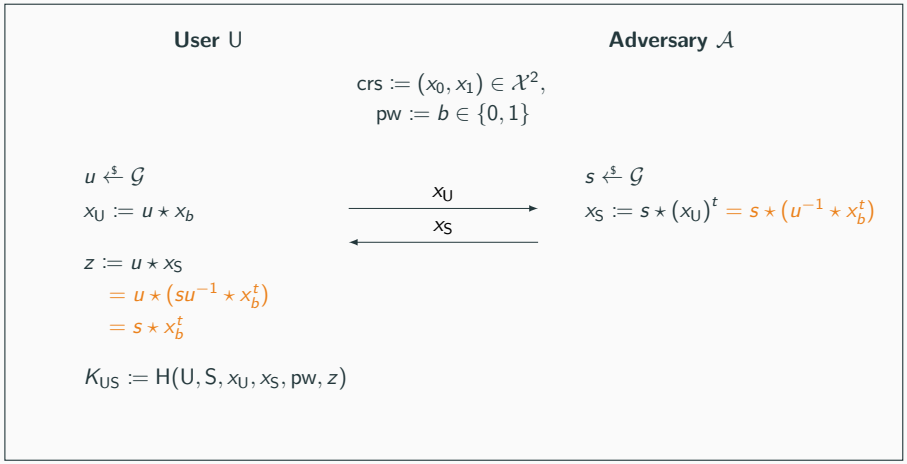
(In)Security of our Protocol

Twists yield an Offline Dictionary Attack!



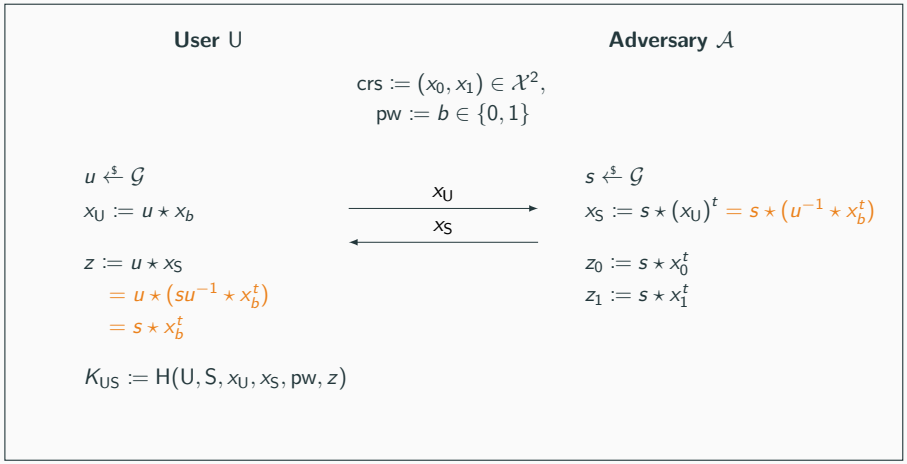
(In)Security of our Protocol

Twists yield an Offline Dictionary Attack!



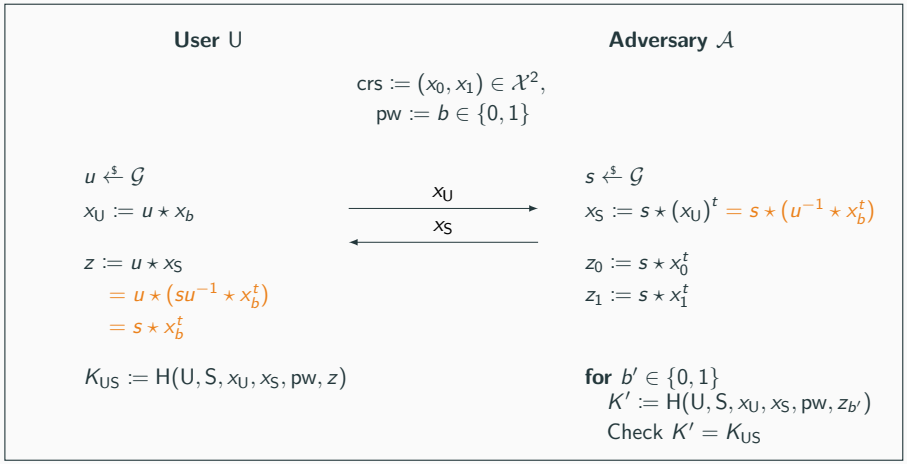
(In)Security of our Protocol

Twists yield an Offline Dictionary Attack!



(In)Security of our Protocol

Twists yield an Offline Dictionary Attack!



Two New PAKE Protocols

Two New PAKE Protocols

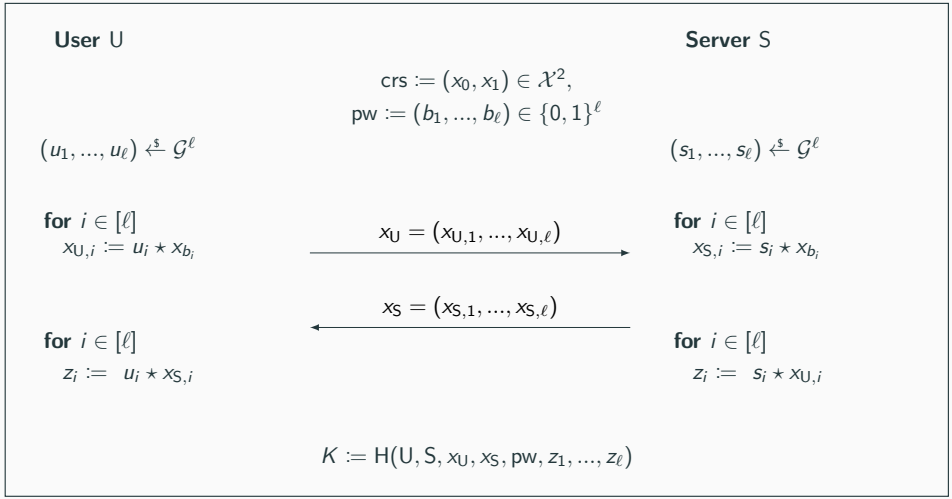
1. **Use a Commitment (**Com-GA-PAKE**)**
   - The server commits on its message using a hash function (random oracle).
   - An adversary cannot choose x_S depending on the user's message.
   - Active security is based on GapCDH.

Two New PAKE Protocols

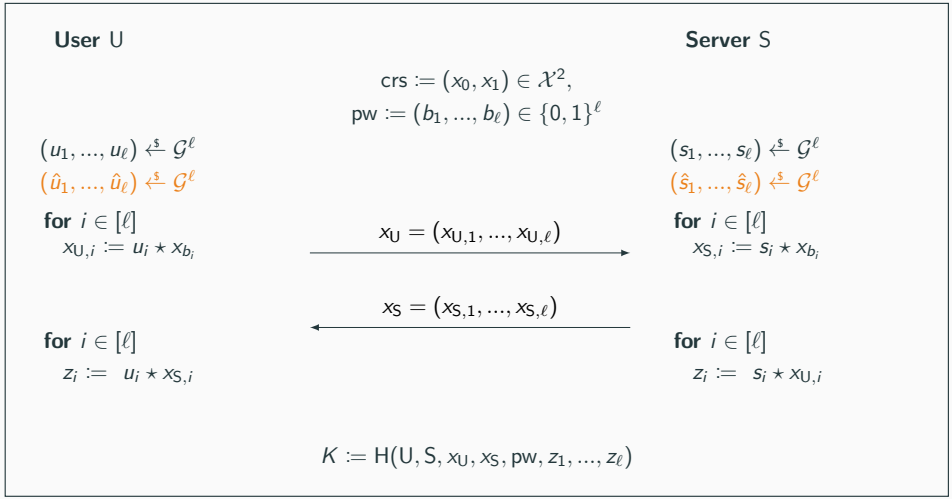
1. **Use a Commitment (**Com-GA-PAKE**)**
   - The server commits on its message using a hash function (random oracle).
   - An adversary cannot choose x_S depending on the user's message.
   - Active security is based on GapCDH.

2. **Use "Cross-Terms" (**X-GA-PAKE**)**
   - Double the communication and combine elements in three ways.
   - $\mathcal{A}$ can compute at most two of the three combinations.

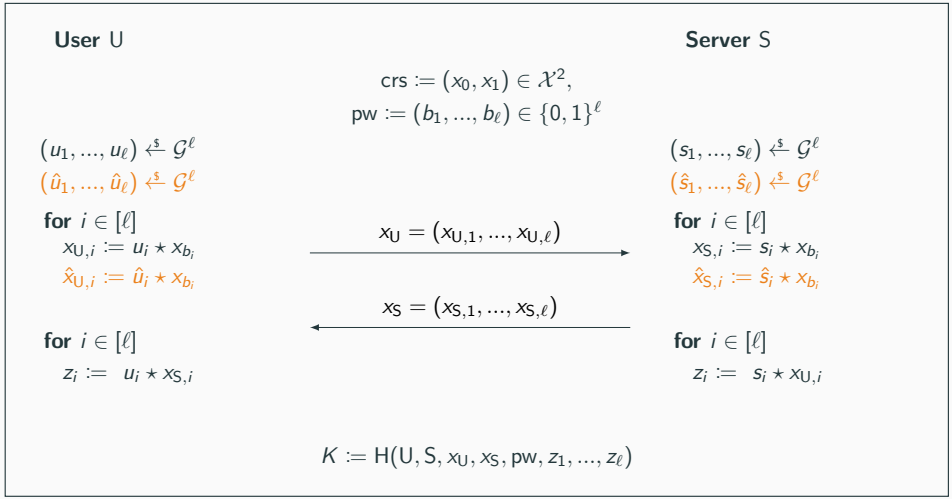
X-GA-PAKE


User U
Server S
crs ≔ $(x_0, x_1) \in \mathcal{X}^2$,
pw ≔ $(b_1, ..., b_\ell) \in \{0,1\}^\ell$
$(u_1, ..., u_\ell) \xleftarrow{\$} \mathcal{G}^\ell$
$(s_1, ..., s_\ell) \xleftarrow{\$} \mathcal{G}^\ell$
for $i \in [\ell]$
$x_{U,i} \coloneqq u_i \star x_{b_i}$
$x_U = (x_{U,1}, ..., x_{U,\ell})$
for $i \in [\ell]$
$x_{S,i} \coloneqq s_i \star x_{b_i}$
$x_S = (x_{S,1}, ..., x_{S,\ell})$
for $i \in [\ell]$
$z_i \coloneqq u_i \star x_{S,i}$
for $i \in [\ell]$
$z_i \coloneqq s_i \star x_{U,i}$
$K \coloneqq \mathsf{H}(\mathsf{U}, \mathsf{S}, x_U, x_S, \mathsf{pw}, z_1, ..., z_\ell)$

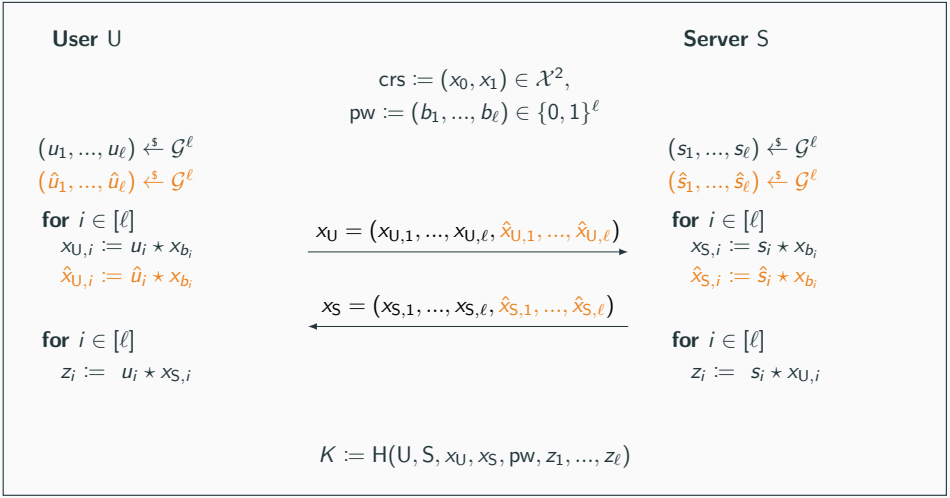
X-GA-PAKE


User U
Server S
crs ≔ (x0, x1) ∈ X^2,
pw ≔ (b1, ..., bℓ) ∈ {0,1}^ℓ
(u1, ..., uℓ) ←$ G^ℓ
(û1, ..., ûℓ) ←$ G^ℓ
(s1, ..., sℓ) ←$ G^ℓ
(ŝ1, ..., ŝℓ) ←$ G^ℓ
for i ∈ [ℓ]
xU,i ≔ ui ⋆ xbi
xU = (xU,1, ..., xU,ℓ)
for i ∈ [ℓ]
xS,i ≔ si ⋆ xbi
xS = (xS,1, ..., xS,ℓ)
for i ∈ [ℓ]
zi ≔ ui ⋆ xS,i
for i ∈ [ℓ]
zi ≔ si ⋆ xU,i
K ≔ H(U, S, xU, xS, pw, z1, ..., zℓ)

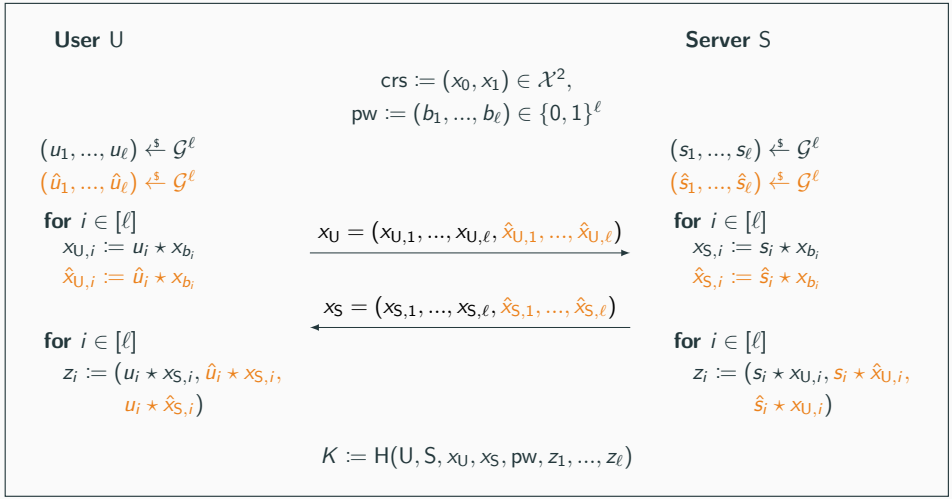
X-GA-PAKE


User U
Server S
crs ≔ $(x_0, x_1) \in \mathcal{X}^2$,
pw ≔ $(b_1, ..., b_\ell) \in \{0,1\}^\ell$
$(u_1, ..., u_\ell) \xleftarrow{\$} \mathcal{G}^\ell$
$(\hat{u}_1, ..., \hat{u}_\ell) \xleftarrow{\$} \mathcal{G}^\ell$
$(s_1, ..., s_\ell) \xleftarrow{\$} \mathcal{G}^\ell$
$(\hat{s}_1, ..., \hat{s}_\ell) \xleftarrow{\$} \mathcal{G}^\ell$
for $i \in [\ell]$
$x_{U,i} \coloneqq u_i \star x_{b_i}$
$\hat{x}_{U,i} \coloneqq \hat{u}_i \star x_{b_i}$
$x_U = (x_{U,1}, ..., x_{U,\ell})$
for $i \in [\ell]$
$x_{S,i} \coloneqq s_i \star x_{b_i}$
$\hat{x}_{S,i} \coloneqq \hat{s}_i \star x_{b_i}$
$x_S = (x_{S,1}, ..., x_{S,\ell})$
for $i \in [\ell]$
$z_i \coloneqq u_i \star x_{S,i}$
for $i \in [\ell]$
$z_i \coloneqq s_i \star x_{U,i}$
$K \coloneqq \mathsf{H}(U, S, x_U, x_S, pw, z_1, ..., z_\ell)$

X-GA-PAKE


User U
Server S
$\mathsf{crs} \coloneqq (x_0, x_1) \in \mathcal{X}^2$,
$\mathsf{pw} \coloneqq (b_1, ..., b_\ell) \in \{0,1\}^\ell$
$(u_1, ..., u_\ell) \xleftarrow{\$} \mathcal{G}^\ell$
$(\hat{u}_1, ..., \hat{u}_\ell) \xleftarrow{\$} \mathcal{G}^\ell$
$(s_1, ..., s_\ell) \xleftarrow{\$} \mathcal{G}^\ell$
$(\hat{s}_1, ..., \hat{s}_\ell) \xleftarrow{\$} \mathcal{G}^\ell$
for $i \in [\ell]$
$x_{\mathsf{U},i} \coloneqq u_i \star x_{b_i}$
$\hat{x}_{\mathsf{U},i} \coloneqq \hat{u}_i \star x_{b_i}$
for $i \in [\ell]$
$x_{\mathsf{S},i} \coloneqq s_i \star x_{b_i}$
$\hat{x}_{\mathsf{S},i} \coloneqq \hat{s}_i \star x_{b_i}$
$x_{\mathsf{U}} = (x_{\mathsf{U},1}, ..., x_{\mathsf{U},\ell}, \hat{x}_{\mathsf{U},1}, ..., \hat{x}_{\mathsf{U},\ell})$
$x_{\mathsf{S}} = (x_{\mathsf{S},1}, ..., x_{\mathsf{S},\ell}, \hat{x}_{\mathsf{S},1}, ..., \hat{x}_{\mathsf{S},\ell})$
for $i \in [\ell]$
$z_i \coloneqq u_i \star x_{\mathsf{S},i}$
for $i \in [\ell]$
$z_i \coloneqq s_i \star x_{\mathsf{U},i}$
$K \coloneqq \mathsf{H}(\mathsf{U}, \mathsf{S}, x_{\mathsf{U}}, x_{\mathsf{S}}, \mathsf{pw}, z_1, ..., z_\ell)$

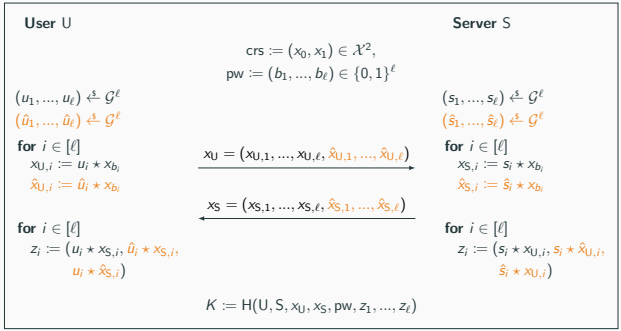
X-GA-PAKE


User U
Server S
crs ≔ (x0, x1) ∈ X^2,
pw ≔ (b1, ..., bℓ) ∈ {0,1}^ℓ
(u1, ..., uℓ) ←$ G^ℓ
(û1, ..., ûℓ) ←$ G^ℓ
(s1, ..., sℓ) ←$ G^ℓ
(ŝ1, ..., ŝℓ) ←$ G^ℓ
for i ∈ [ℓ]
xU,i ≔ ui ⋆ xbi
x̂U,i ≔ ûi ⋆ xbi
for i ∈ [ℓ]
xS,i ≔ si ⋆ xbi
x̂S,i ≔ ŝi ⋆ xbi
xU = (xU,1, ..., xU,ℓ, x̂U,1, ..., x̂U,ℓ)
xS = (xS,1, ..., xS,ℓ, x̂S,1, ..., x̂S,ℓ)
for i ∈ [ℓ]
zi ≔ (ui ⋆ xS,i, ûi ⋆ xS,i, ui ⋆ x̂S,i)
for i ∈ [ℓ]
zi ≔ (si ⋆ xU,i, si ⋆ x̂U,i, ŝi ⋆ xU,i)
K ≔ H(U, S, xU, xS, pw, z1, ..., zℓ)

Security of X-GA-PAKE

Security against Passive Adversaries

- secure under Strong CDH + ROM



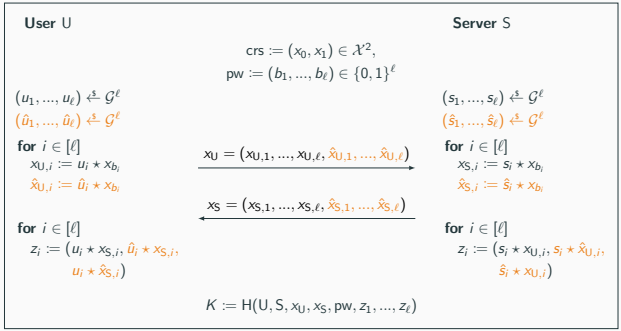
Security of X-GA-PAKE

Security against Passive Adversaries

- secure under Strong CDH + ROM

Security against Active Adversaries

- secure under Strong Square-Inverse DH + ROM



Security of X-GA-PAKE

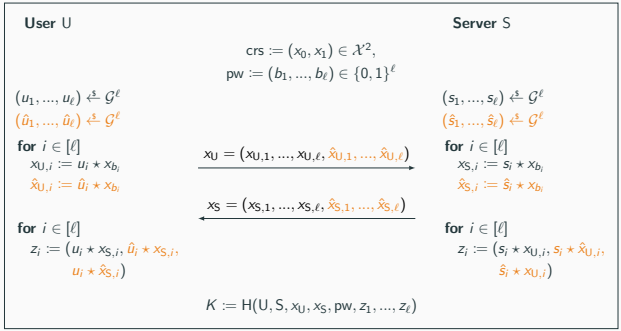
Security against Passive Adversaries

- secure under Strong CDH + ROM

Security against Active Adversaries

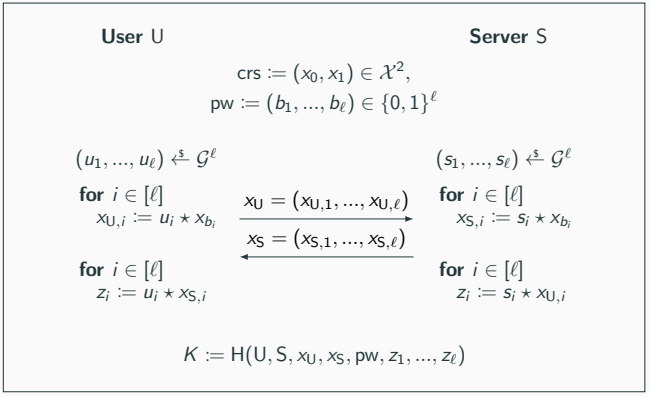
- secure under Strong Square-Inverse DH + ROM
- given $(g \star \tilde{x})$ compute (y, z_0, z_1) such that

$$z_0 = g^2 \star y$$
$$z_1 = g^{-1} \star y$$

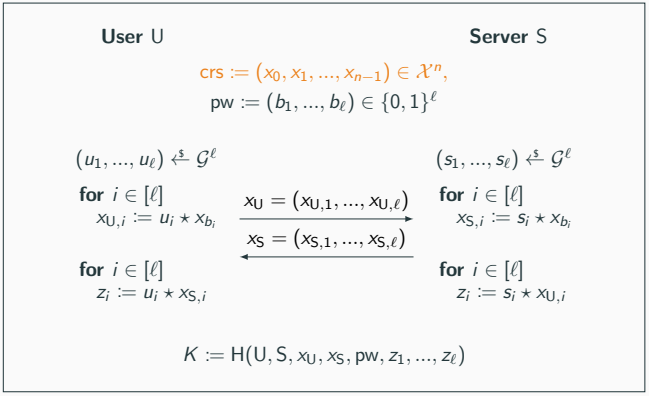


Optimizations

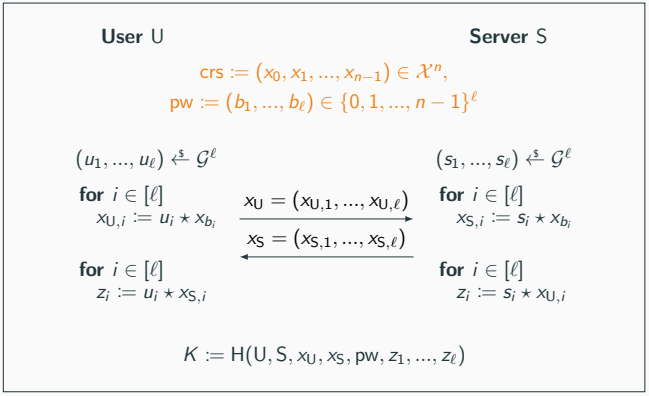
Decrease the Communication and Computation Cost



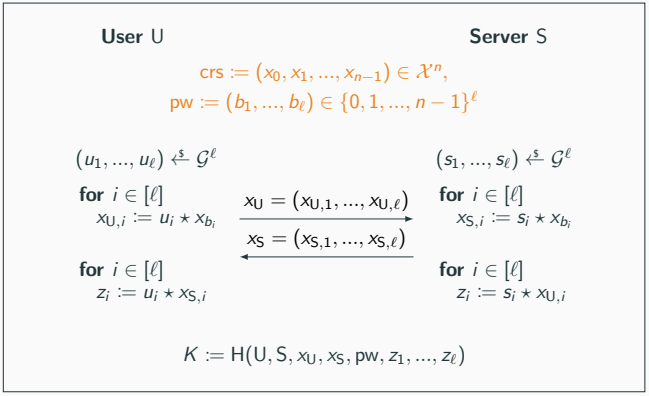
Decrease the Communication and Computation Cost



Decrease the Communication and Computation Cost

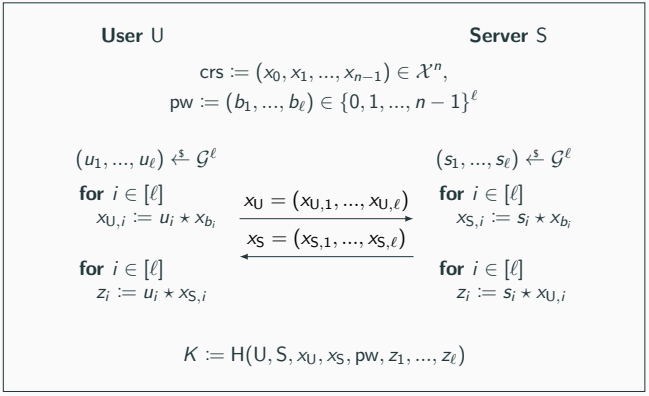


Decrease the Communication and Computation Cost

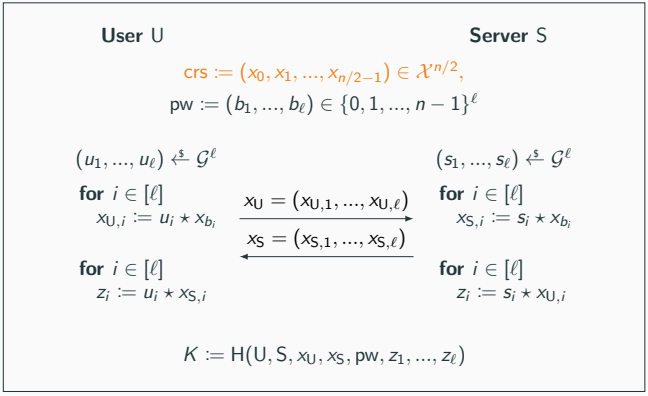


$\Rightarrow$ By increasing the number of public parameters n, we can choose a smaller ℓ.

Use Twists in the Setup



Use Twists in the Setup



Here we implicitly define $x_{n/2+i} := x_i^t$ for $i \in \{0, 1, ..., n-1\}$.

Comparison of Com-GA-PAKE and X-GA-PAKE

	Using OT [CDVW12, LGd21]	Com-GA-PAKE	X-GA-PAKE
Set Elements	384	16	32
Evaluations	1408	32	80
Rounds	4	3	1
Security Assumption	CDH	Gap CDH	Strong Square-Inverse
Tight	no	no	yes

Here we assume $\mathcal{PW} \subset \{0,1\}^{128}$ for all schemes, i.e.,

- $\ell = 128$ for the OT-based construction
- $\ell = 16$, $n = 8$ for our optimized variants

Comparison of Com-GA-PAKE and X-GA-PAKE

	Using OT [CDVW12, LGd21]	Com-GA-PAKE	X-GA-PAKE
Set Elements	384	16	32
Evaluations	1408	32	80
Rounds	4	3	1
Security Assumption	CDH	Gap CDH	Strong Square-Inverse
Tight	no	no	yes

Here we assume $\mathcal{PW} \subset \{0,1\}^{128}$ for all schemes, i.e.,

- $\ell = 128$ for the OT-based construction
- $\ell = 16$, $n = 8$ for our optimized variants

Comparison of Com-GA-PAKE and X-GA-PAKE

	Using OT [CDVW12, LGd21]	Com-GA-PAKE	X-GA-PAKE
Set Elements	384	16	32
Evaluations	1408	32	80
Rounds	4	3	1
Security Assumption	CDH	Gap CDH	Strong Square-Inverse
Tight	no	no	yes

Here we assume $\mathcal{PW} \subset \{0,1\}^{128}$ for all schemes, i.e.,

- $\ell = 128$ for the OT-based construction
- $\ell = 16$, $n = 8$ for our optimized variants

Comparison of Com-GA-PAKE and X-GA-PAKE

	Using OT [CDVW12, LGd21]	Com-GA-PAKE	X-GA-PAKE
Set Elements	384	16	32
Evaluations	1408	32	80
Rounds	4	3	1
Security Assumption	CDH	Gap CDH	Strong Square-Inverse
Tight	no	no	yes

Here we assume $\mathcal{PW} \subset \{0,1\}^{128}$ for all schemes, i.e.,

- $\ell = 128$ for the OT-based construction
- $\ell = 16$, $n = 8$ for our optimized variants

Comparison of Com-GA-PAKE and X-GA-PAKE

	Using OT [CDVW12, LGd21]	Com-GA-PAKE	X-GA-PAKE
Set Elements	384	16	32
Evaluations	1408	32	80
Rounds	4	3	1
Security Assumption	CDH	Gap CDH	Strong Square-Inverse
Tight	no	no	yes

Here we assume $\mathcal{PW} \subset \{0,1\}^{128}$ for all schemes, i.e.,

- $\ell = 128$ for the OT-based construction
- $\ell = 16$, $n = 8$ for our optimized variants

Comparison of Com-GA-PAKE and X-GA-PAKE

	Using OT [CDVW12, LGd21]	Com-GA-PAKE	X-GA-PAKE
Set Elements	384	16	32
Evaluations	1408	32	80
Rounds	4	3	1
Security Assumption	CDH	Gap CDH	Strong Square-Inverse
Tight	no	no	yes

Here we assume $\mathcal{PW} \subset \{0,1\}^{128}$ for all schemes, i.e.,

- $\ell = 128$ for the OT-based construction
- $\ell = 16$, $n = 8$ for our optimized variants

Summary

Results

- Group actions with twists as abstraction for CSIDH
- The first direct constructions and provably secure PAKE protocols from CSIDH
- Better efficiency than generic constructions, e.g. based on OT

Summary

Results

- Group actions with twists as abstraction for CSIDH
- The first direct constructions and provably secure PAKE protocols from CSIDH
- Better efficiency than generic constructions, e.g. based on OT

Thank you!

ePrint: ia.cr/2022/770 doreen.riepel@rub.de

References i

Navid Alamati, Luca De Feo, Hart Montgomery, and Sikhar Patranabis.
Cryptographic group actions and applications.
In Shiho Moriai and Huaxiong Wang, editors, ASIACRYPT 2020, Part II, volume 12492 of LNCS, pages 411–439. Springer, Heidelberg, December 2020.

Reza Azarderakhsh, David Jao, Brian Koziel, Jason T. LeGrow, Vladimir Soukharev, and Oleg Taraskin.
How not to create an isogeny-based PAKE.
In Mauro Conti, Jianying Zhou, Emiliano Casalicchio, and Angelo Spognardi, editors, ACNS 20, Part I, volume 12146 of LNCS, pages 169–186. Springer, Heidelberg, October 2020.

References ii

Jeremy Booher, Ross Bowden, Javad Doliskani, Tako Boris Fouotsa, Steven D. Galbraith, Sabrina Kunzweiler, Simon-Philipp Merz, Christophe Petit, Benjamin Smith, Katherine E. Stange, Yan Bo Ti, Christelle Vincent, José Felipe Voloch, Charlotte Weitkämper, and Lukas Zobernig.

Failing to hash into supersingular isogeny graphs.

Cryptology ePrint Archive, Report 2022/518, 2022.

https://eprint.iacr.org/2022/518.

Ran Canetti, Dana Dachman-Soled, Vinod Vaikuntanathan, and Hoeteck Wee.

Efficient password authenticated key exchange via oblivious transfer.

In Marc Fischlin, Johannes Buchmann, and Mark Manulis, editors, PKC 2012, volume 7293 of LNCS, pages 449–466. Springer, Heidelberg, May 2012.

References iii


Wouter Castryck, Tanja Lange, Chloe Martindale, Lorenz Panny, and Joost Renes.
CSIDH: An efficient post-quantum commutative group action.
In Thomas Peyrin and Steven Galbraith, editors, ASIACRYPT 2018, Part III, volume 11274 of LNCS, pages 395–427. Springer, Heidelberg, December 2018.


David P Jablon.
Strong password-only authenticated key exchange.
ACM SIGCOMM Computer Communication Review, 26(5):5–26, 1996.


Yi-Fu Lai, Steven D. Galbraith, and Cyprien de Saint Guilhem.
Compact, efficient and UC-secure isogeny-based oblivious transfer.
In Anne Canteaut and François-Xavier Standaert, editors, EUROCRYPT 2021, Part I, volume 12696 of LNCS, pages 213–241. Springer, Heidelberg, October 2021.

References iv


Marzio Mula, Nadir Murru, and Federico Pintore.
On random sampling of supersingular elliptic curves.
Cryptology ePrint Archive, Paper 2022/528, 2022.
https://eprint.iacr.org/2022/528.