Hardware Operators for Pairing-Based Cryptography

— Part I: Because size matters —

Jean-Luc Beuchat

Laboratory of Cryptography and Informantion Security
University of Tsukuba, Japan
jeanluc.beuchat@gmail.com

Joint work with:

Nicolas Brisebarre	Arénaire, LIP, ÉNS Lyon, France
Jérémie Detrey	CACAO, LORIA, Nancy, France
Nicolas Estibals	CACAO, LORIA, Nancy, France
Eiji Okamoto	LCIS, University of Tsukuba, Japan
Francisco Rodríguez-Henríquez	CSD, IPN, Mexico City, Mexico

Outline of the talk

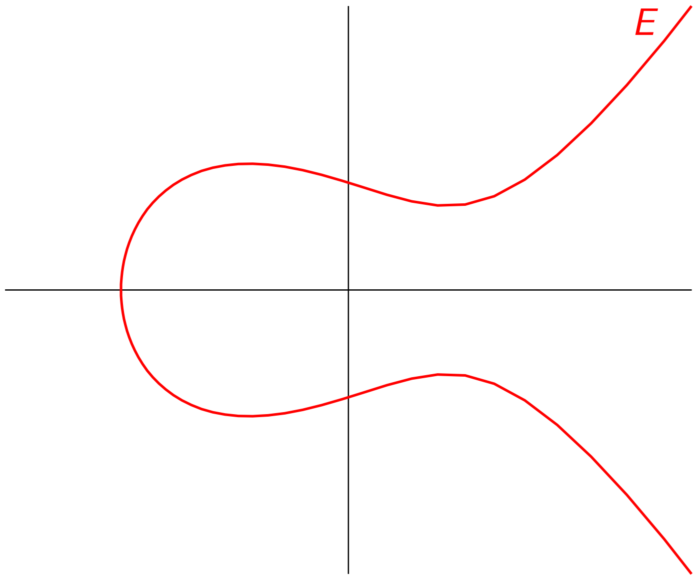
- Pairing-based cryptography
- Pairings over elliptic curves
- Finite-field arithmetic
- Implementation results
- Concluding thoughts

Outline of the talk

- Pairing-based cryptography
- Pairings over elliptic curves
- Finite-field arithmetic
- Implementation results
- Concluding thoughts

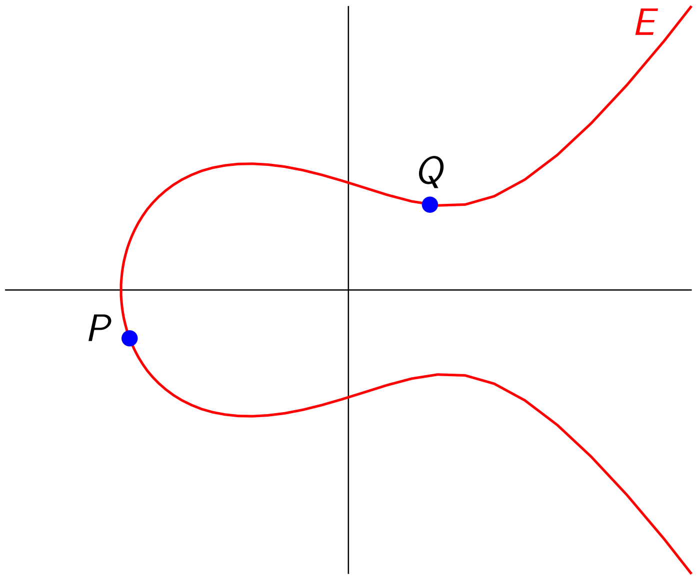
Elliptic curves

- E defined by a Weierstraß equation of the form $y^2 = x^3 + Ax + B$



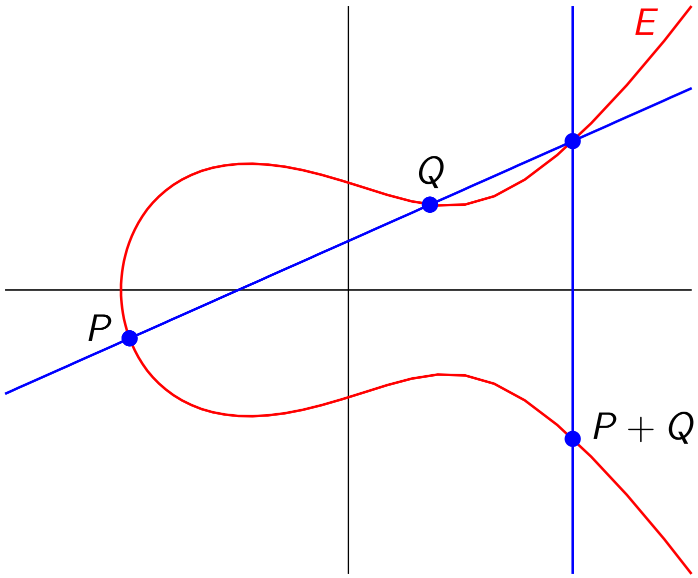
Elliptic curves

- E defined by a Weierstraß equation of the form $y^2 = x^3 + Ax + B$
- $E(K)$ set of rational points over a field K



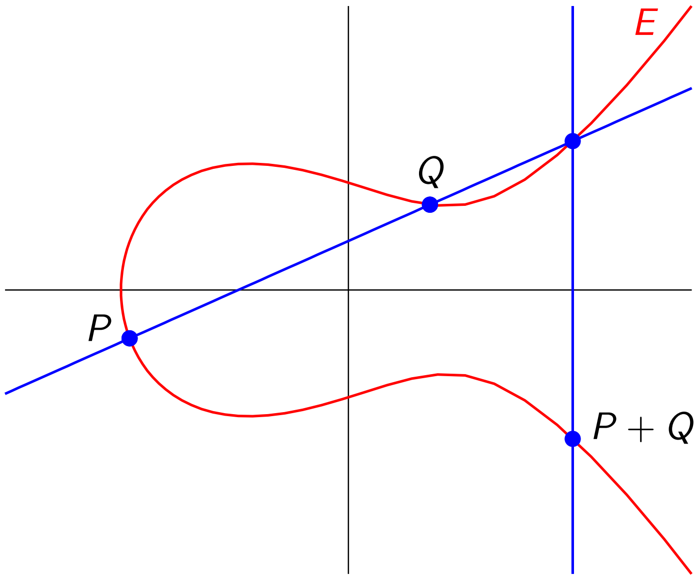
Elliptic curves

- E defined by a Weierstraß equation of the form $y^2 = x^3 + Ax + B$
- $E(K)$ set of rational points over a field K
- Additive group law over $E(K)$



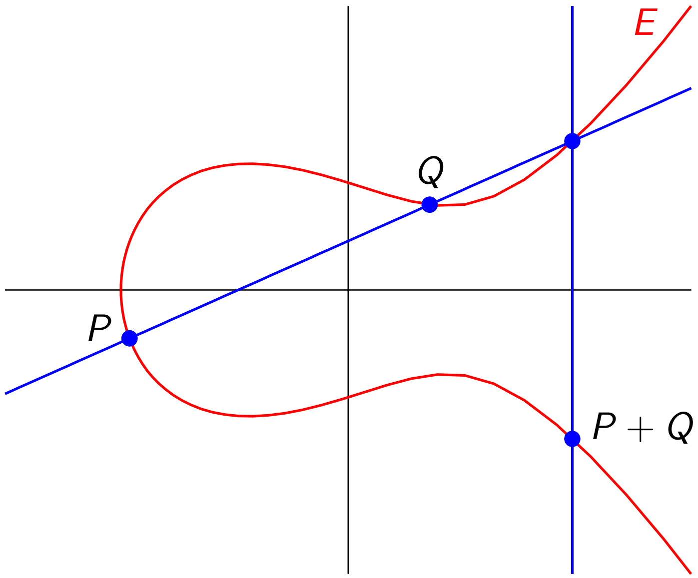
Elliptic curves

- E defined by a Weierstraß equation of the form
  $y^2 = x^3 + Ax + B$
- $E(K)$ set of rational points over a field K
- Additive group law over $E(K)$
- Many applications in cryptography since 1985
  - EC-based Diffie-Hellman key exchange
  - EC-based Digital Signature Algorithm
  - ...
- Interest: smaller keys than usual cryptosystems (RSA, DSA, ElGamal, ...)



Elliptic curves

- E defined by a Weierstraß equation of the form $y^2 = x^3 + Ax + B$
- $E(K)$ set of rational points over a field K
- Additive group law over $E(K)$
- Many applications in cryptography since 1985
  - EC-based Diffie-Hellman key exchange
  - EC-based Digital Signature Algorithm
  - ...
- Interest: smaller keys than usual cryptosystems (RSA, DSA, ElGamal, ...)
- But there's more: bilinear pairings

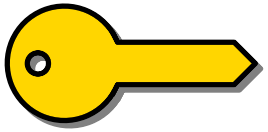


Group cryptography

- $(\mathbb{G}_1, +)$, an additively-written cyclic group of prime order $\#\mathbb{G}_1 = \ell$

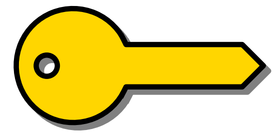
Group cryptography

- $(\mathbb{G}_1, +)$, an additively-written cyclic group of prime order $\#\mathbb{G}_1 = \ell$
- P, a generator of the group: $\mathbb{G}_1 = \langle P \rangle$



Group cryptography

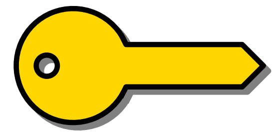
- $(\mathbb{G}_1, +)$, an additively-written cyclic group of prime order $\#\mathbb{G}_1 = \ell$
- P, a generator of the group: $\mathbb{G}_1 = \langle P \rangle$



- Scalar multiplication: for any integer k, we have $kP = \underbrace{P + P + \cdots + P}_{k \text{ times}}$

Group cryptography

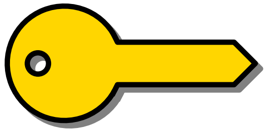
- $(\mathbb{G}_1, +)$, an additively-written cyclic group of prime order $\#\mathbb{G}_1 = \ell$
- P, a generator of the group: $\mathbb{G}_1 = \langle P \rangle$



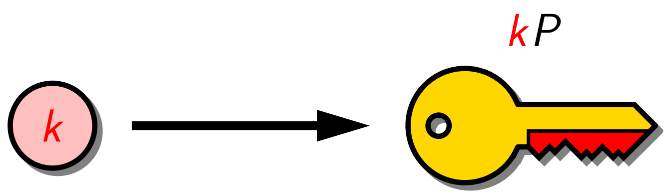
- Scalar multiplication: for any integer k, we have $kP = \underbrace{P + P + \cdots + P}_{k \text{ times}}$

Group cryptography

- $(\mathbb{G}_1, +)$, an additively-written cyclic group of prime order $\#\mathbb{G}_1 = \ell$
- P, a generator of the group: $\mathbb{G}_1 = \langle P \rangle$

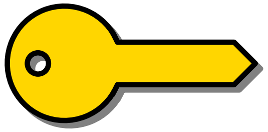


- Scalar multiplication: for any integer k, we have $kP = \underbrace{P + P + \cdots + P}_{k \text{ times}}$

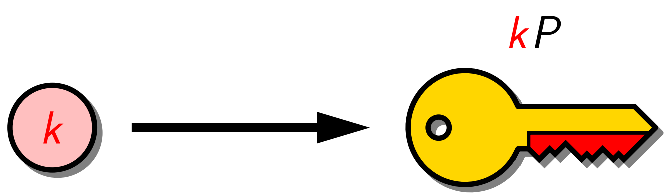


Group cryptography

- $(\mathbb{G}_1, +)$, an additively-written cyclic group of prime order $\#\mathbb{G}_1 = \ell$
- P, a generator of the group: $\mathbb{G}_1 = \langle P \rangle$



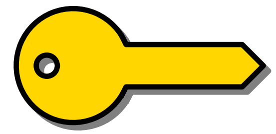
- Scalar multiplication: for any integer k, we have $kP = \underbrace{P + P + \cdots + P}_{k \text{ times}}$



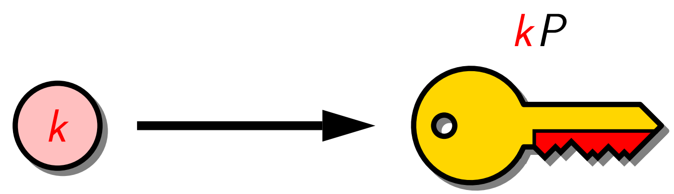
- Discrete logarithm: given $Q \in \mathbb{G}_1$, compute k such that $Q = kP$

Group cryptography

- $(\mathbb{G}_1, +)$, an additively-written cyclic group of prime order $\#\mathbb{G}_1 = \ell$
- P, a generator of the group: $\mathbb{G}_1 = \langle P \rangle$

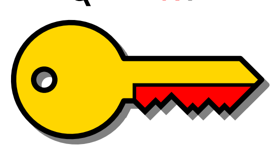


- Scalar multiplication: for any integer k, we have $kP = \underbrace{P + P + \cdots + P}_{k \text{ times}}$



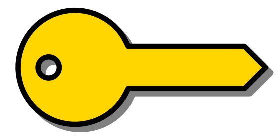
- Discrete logarithm: given $Q \in \mathbb{G}_1$, compute k such that $Q = kP$

$Q = kP$

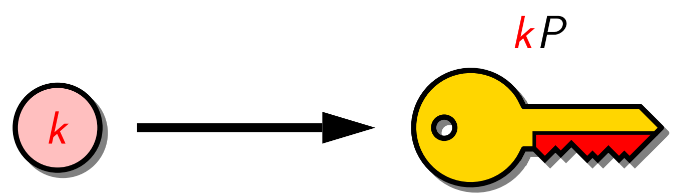


Group cryptography

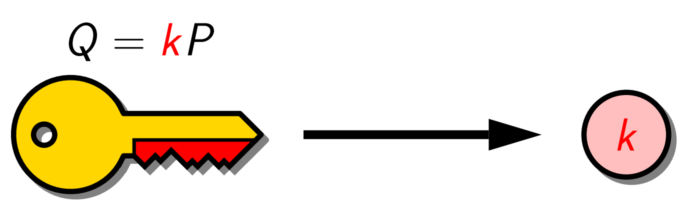
- $(\mathbb{G}_1, +)$, an additively-written cyclic group of prime order $\#\mathbb{G}_1 = \ell$
- P, a generator of the group: $\mathbb{G}_1 = \langle P \rangle$



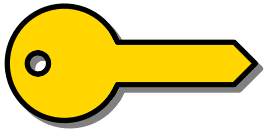
- Scalar multiplication: for any integer k, we have $kP = \underbrace{P + P + \cdots + P}_{k \text{ times}}$

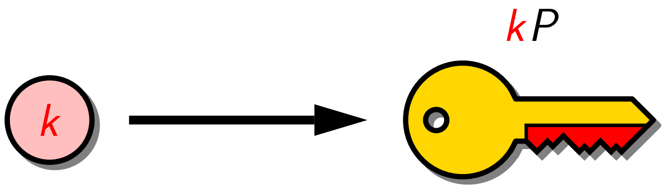


- Discrete logarithm: given $Q \in \mathbb{G}_1$, compute k such that $Q = kP$

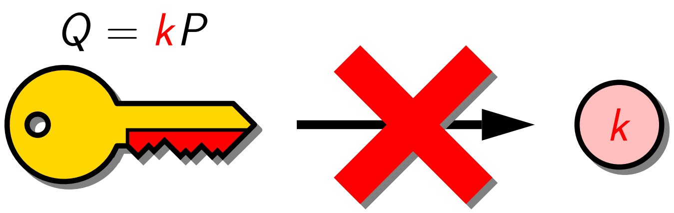


Group cryptography

- $(\mathbb{G}_1, +)$, an additively-written cyclic group of prime order $\#\mathbb{G}_1 = \ell$
- P, a generator of the group: $\mathbb{G}_1 = \langle P \rangle$ 
- Scalar multiplication: for any integer k, we have $kP = \underbrace{P + P + \cdots + P}_{k \text{ times}}$



- Discrete logarithm: given $Q \in \mathbb{G}_1$, compute k such that $Q = kP$



- We assume that the discrete logarithm problem (DLP) in $\mathbb{G}_1$ is hard

Bilinear pairings

- $(\mathbb{G}_2, \times)$, a multiplicatively-written cyclic group of order $\#\mathbb{G}_2 = \#\mathbb{G}_1 = \ell$

Bilinear pairings

- $(\mathbb{G}_2, \times)$, a multiplicatively-written cyclic group of order $\#\mathbb{G}_2 = \#\mathbb{G}_1 = \ell$
- A bilinear pairing on $(\mathbb{G}_1, \mathbb{G}_2)$ is a map

$$\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$$

  that satisfies the following conditions:

  - non-degeneracy: $\hat{e}(P, P) \neq 1_{\mathbb{G}_2}$ (equivalently $\hat{e}(P, P)$ generates $\mathbb{G}_2$)
  - bilinearity:

$$\hat{e}(Q_1 + Q_2, R) = \hat{e}(Q_1, R) \cdot \hat{e}(Q_2, R) \qquad \hat{e}(Q, R_1 + R_2) = \hat{e}(Q, R_1) \cdot \hat{e}(Q, R_2)$$

  - computability: $\hat{e}$ can be efficiently computed

Bilinear pairings

- $(\mathbb{G}_2, \times)$, a multiplicatively-written cyclic group of order $\#\mathbb{G}_2 = \#\mathbb{G}_1 = \ell$
- A bilinear pairing on $(\mathbb{G}_1, \mathbb{G}_2)$ is a map
$$\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$$
that satisfies the following conditions:
  - non-degeneracy: $\hat{e}(P, P) \neq 1_{\mathbb{G}_2}$ (equivalently $\hat{e}(P, P)$ generates $\mathbb{G}_2$)
  - bilinearity:
$$\hat{e}(Q_1 + Q_2, R) = \hat{e}(Q_1, R) \cdot \hat{e}(Q_2, R) \qquad \hat{e}(Q, R_1 + R_2) = \hat{e}(Q, R_1) \cdot \hat{e}(Q, R_2)$$
  - computability: $\hat{e}$ can be efficiently computed
- Immediate property: for any two integers k_1 and k_2
$$\hat{e}(k_1 Q, k_2 R) = \hat{e}(Q, R)^{k_1 k_2}$$

Bilinear pairings

- $(\mathbb{G}_2, \times)$, a multiplicatively-written cyclic group of order $\#\mathbb{G}_2 = \#\mathbb{G}_1 = \ell$
- A bilinear pairing on $(\mathbb{G}_1, \mathbb{G}_2)$ is a map

  $$\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$$

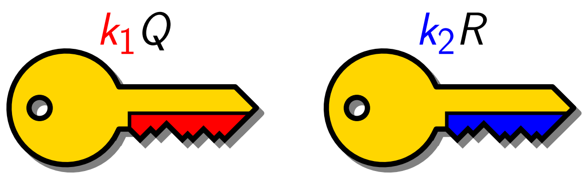
  that satisfies the following conditions:

  - non-degeneracy: $\hat{e}(P, P) \neq 1_{\mathbb{G}_2}$ (equivalently $\hat{e}(P, P)$ generates $\mathbb{G}_2$)
  - bilinearity:

    $$\hat{e}(Q_1 + Q_2, R) = \hat{e}(Q_1, R) \cdot \hat{e}(Q_2, R) \qquad \hat{e}(Q, R_1 + R_2) = \hat{e}(Q, R_1) \cdot \hat{e}(Q, R_2)$$

  - computability: $\hat{e}$ can be efficiently computed
- Immediate property: for any two integers k_1 and k_2

  $$\hat{e}(k_1 Q, k_2 R) = \hat{e}(Q, R)^{k_1 k_2}$$



Bilinear pairings

- $(\mathbb{G}_2, \times)$, a multiplicatively-written cyclic group of order $\#\mathbb{G}_2 = \#\mathbb{G}_1 = \ell$
- A bilinear pairing on $(\mathbb{G}_1, \mathbb{G}_2)$ is a map

$$\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$$

  that satisfies the following conditions:

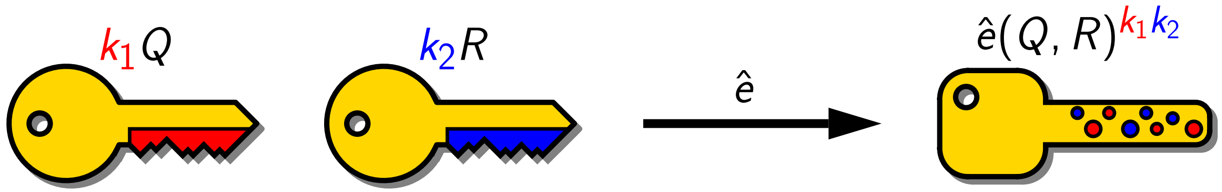
  - non-degeneracy: $\hat{e}(P, P) \neq 1_{\mathbb{G}_2}$ (equivalently $\hat{e}(P, P)$ generates $\mathbb{G}_2$)
  - bilinearity:

$$\hat{e}(Q_1 + Q_2, R) = \hat{e}(Q_1, R) \cdot \hat{e}(Q_2, R) \qquad \hat{e}(Q, R_1 + R_2) = \hat{e}(Q, R_1) \cdot \hat{e}(Q, R_2)$$

  - computability: $\hat{e}$ can be efficiently computed

- Immediate property: for any two integers k_1 and k_2

$$\hat{e}(k_1 Q, k_2 R) = \hat{e}(Q, R)^{k_1 k_2}$$



Pairings in cryptography

- At first, used to attack supersingular elliptic curves
  - Menezes-Okamoto-Vanstone and Frey-Rück attacks, 1993 and 1994

$$\mathrm{DLP}_{\mathbb{G}_1}$$

$$kP$$

Pairings in cryptography

- At first, used to attack supersingular elliptic curves
  - Menezes-Okamoto-Vanstone and Frey-Rück attacks, 1993 and 1994

$$\mathrm{DLP}_{\mathbb{G}_1} \quad <_{\mathrm{P}} \quad \mathrm{DLP}_{\mathbb{G}_2}$$

$$kP \quad \longrightarrow \quad \hat{e}(kP, P) = \hat{e}(P, P)^k$$

Pairings in cryptography

- At first, used to attack supersingular elliptic curves
  - Menezes-Okamoto-Vanstone and Frey-Rück attacks, 1993 and 1994

$$\mathrm{DLP}_{\mathbb{G}_1} \quad <_{\mathrm{P}} \quad \mathrm{DLP}_{\mathbb{G}_2}$$

$$kP \quad \longrightarrow \quad \hat{e}(kP, P) = \hat{e}(P, P)^k$$

  - for cryptographic applications, we will also require the DLP in $\mathbb{G}_2$ to be hard

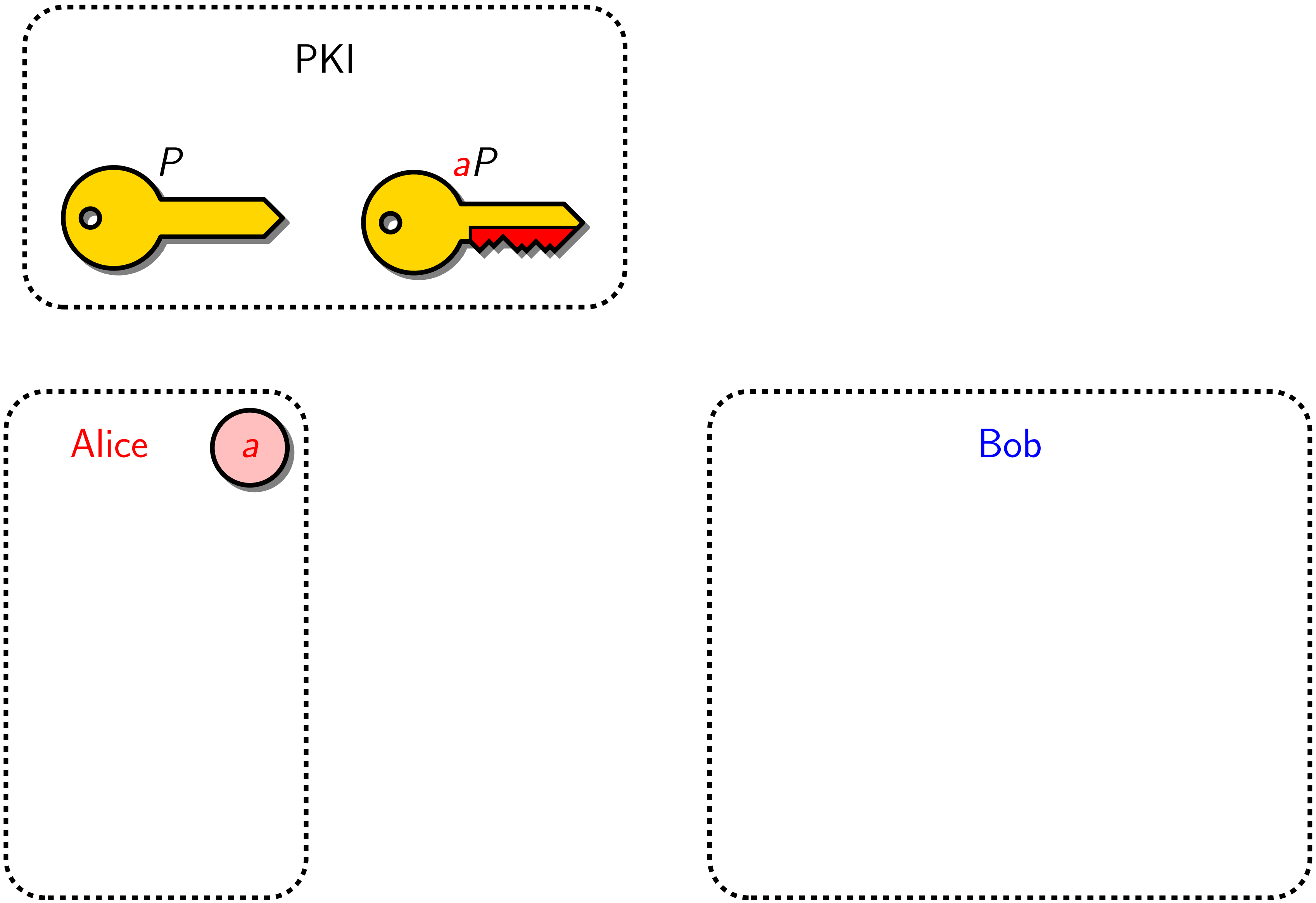
Pairings in cryptography

- At first, used to attack supersingular elliptic curves
  - Menezes-Okamoto-Vanstone and Frey-Rück attacks, 1993 and 1994

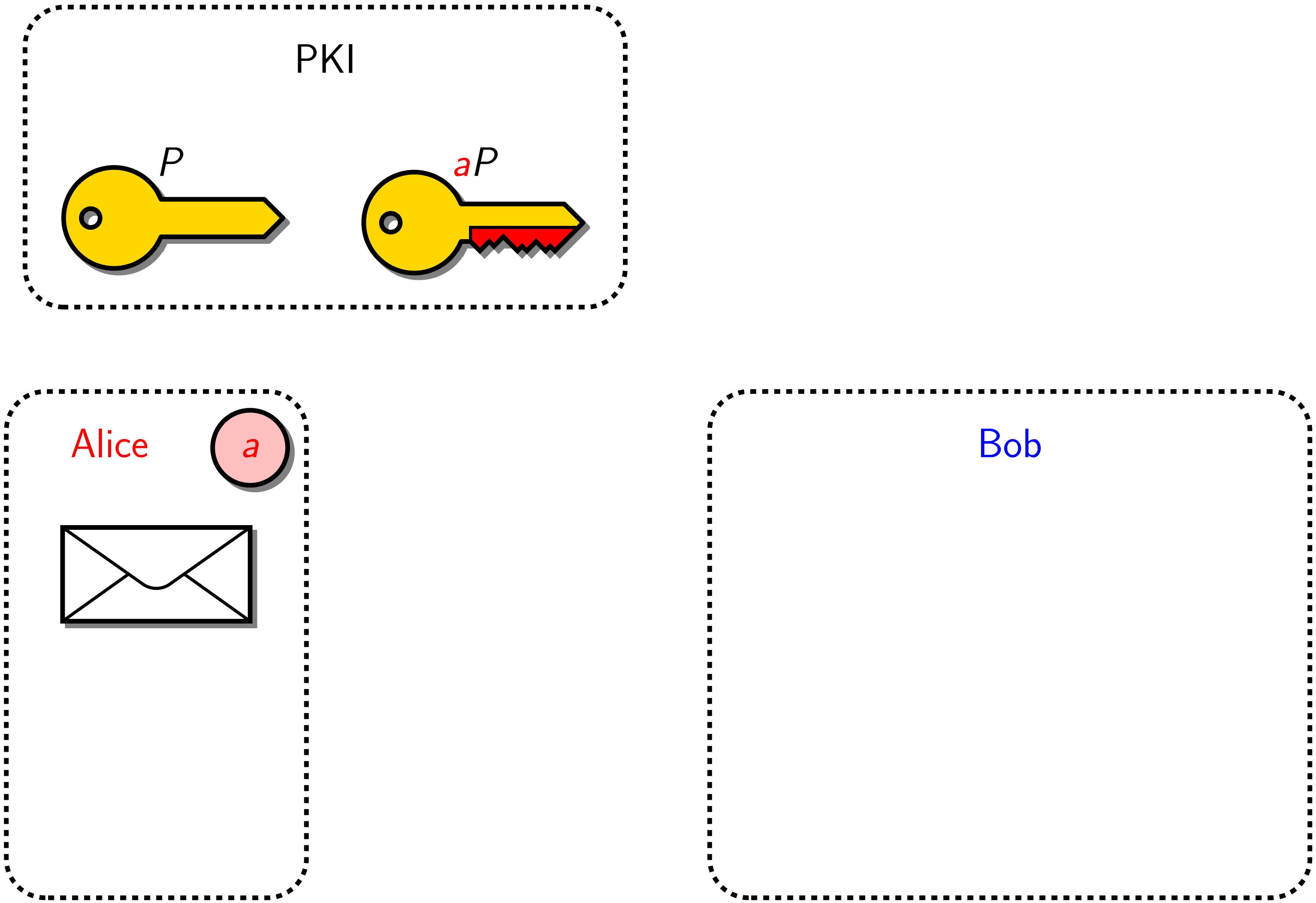
$$\mathrm{DLP}_{\mathbb{G}_1} \quad <_{\mathrm{P}} \quad \mathrm{DLP}_{\mathbb{G}_2}$$
$$kP \quad \longrightarrow \quad \hat{e}(kP, P) = \hat{e}(P, P)^k$$

  - for cryptographic applications, we will also require the DLP in $\mathbb{G}_2$ to be hard
- One-round three-party key agreement (Joux, 2000)
- Identity-based encryption
  - Boneh-Franklin, 2001
  - Sakai-Kasahara, 2001
- Short digital signatures
  - Boneh-Lynn-Shacham, 2001
  - Zang-Safavi-Naini-Susilo, 2004
- ...

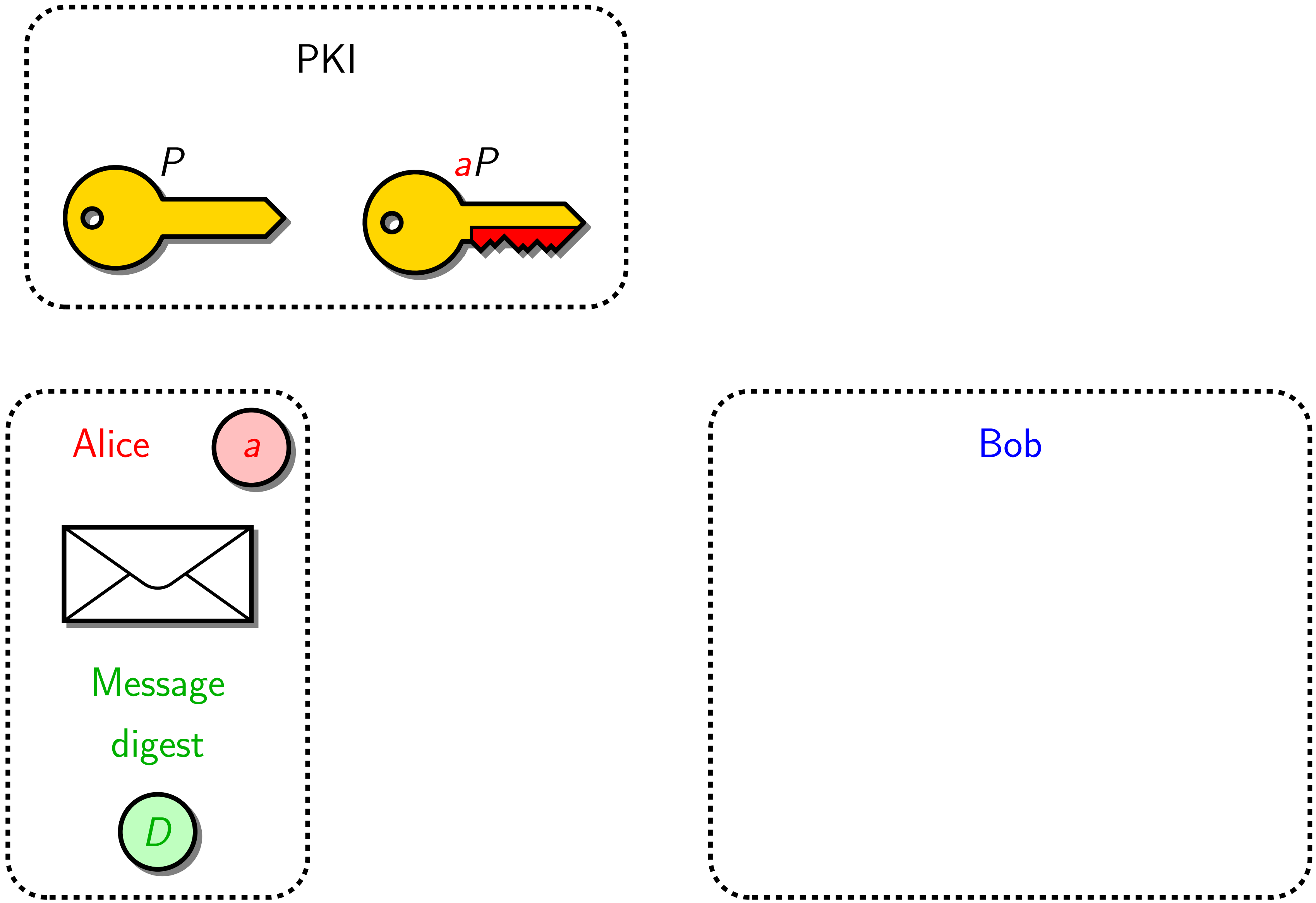
Short signature (Boneh, Lynn & Shacham, 2001)



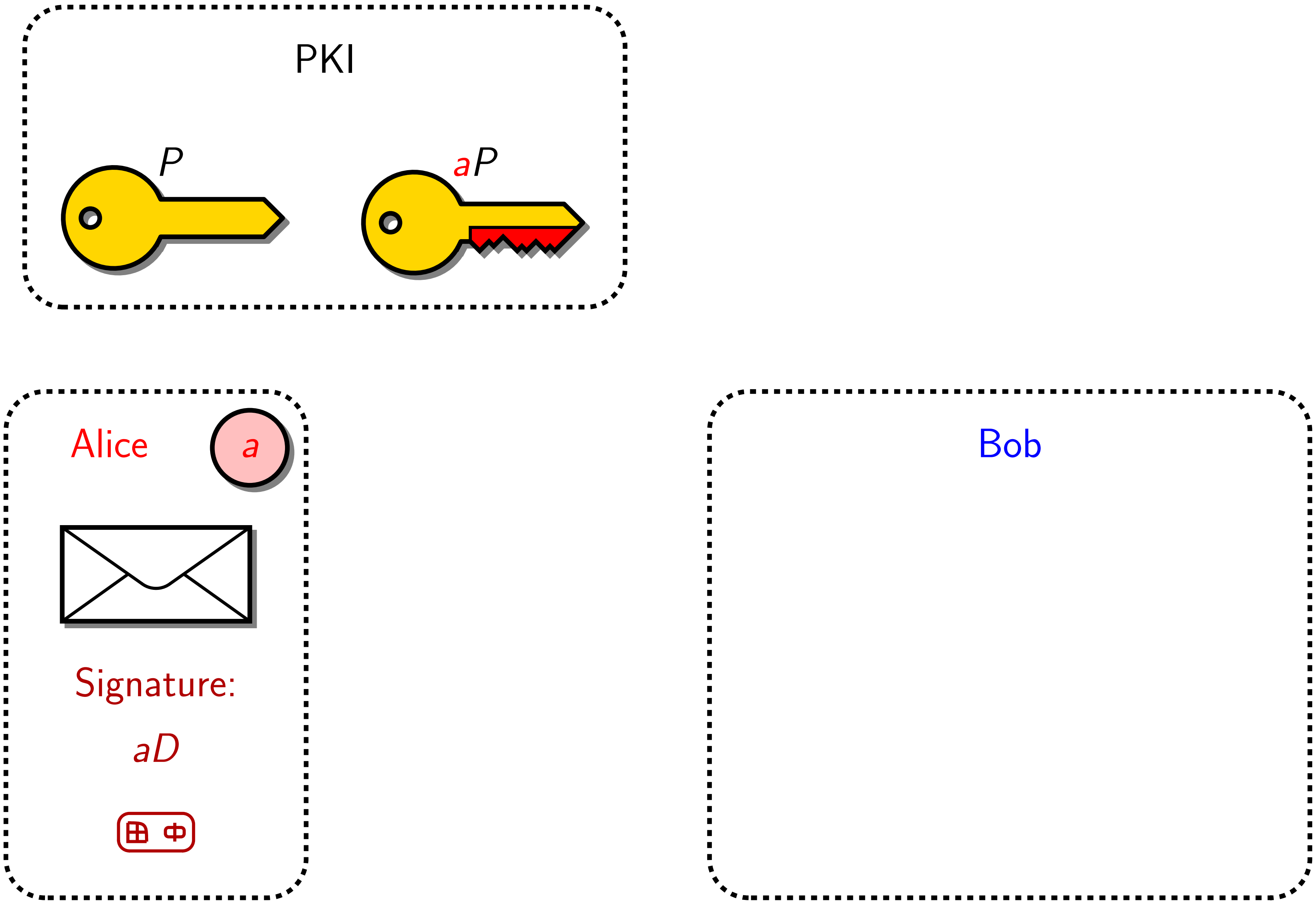
Short signature (Boneh, Lynn & Shacham, 2001)



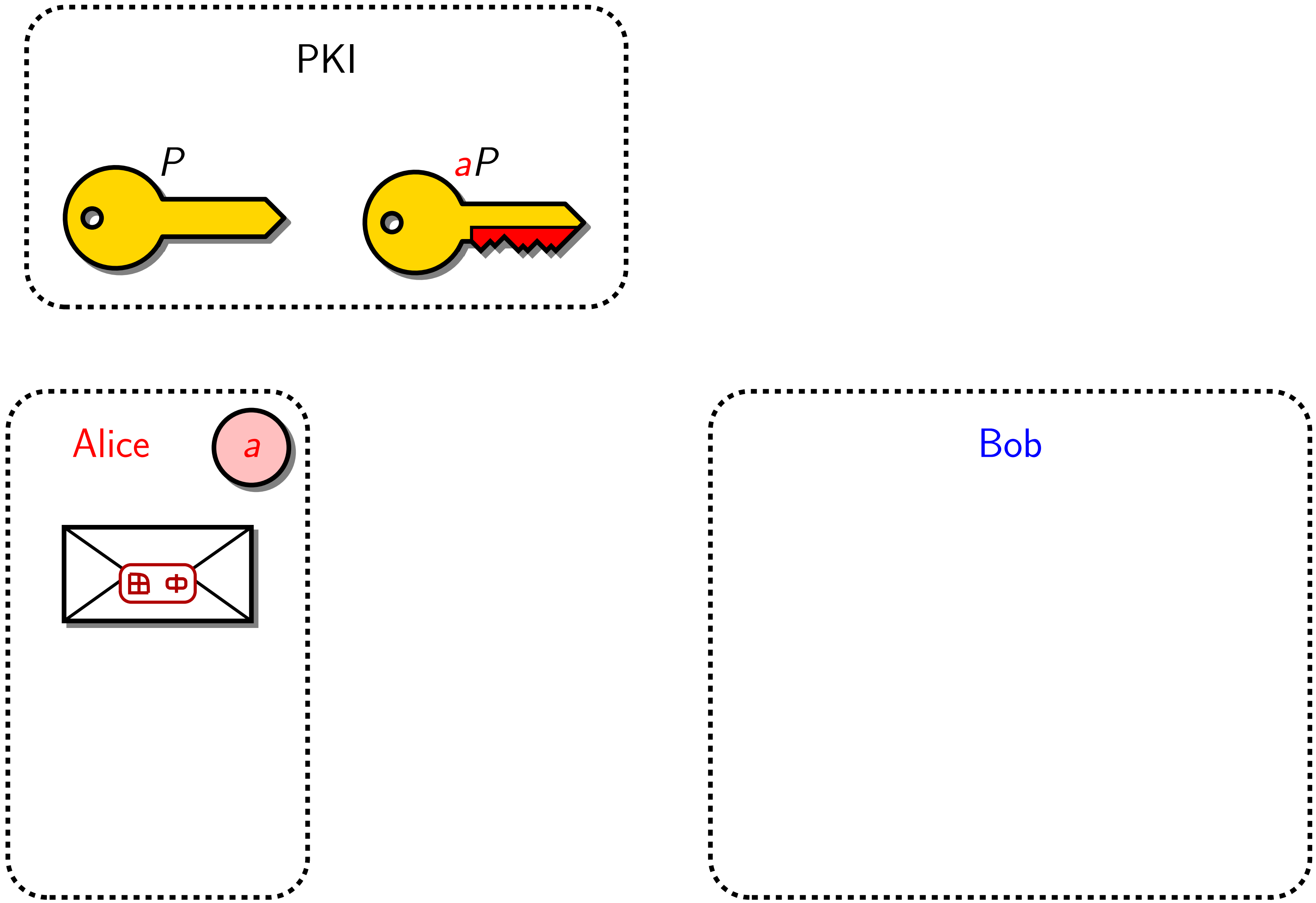
Short signature (Boneh, Lynn & Shacham, 2001)



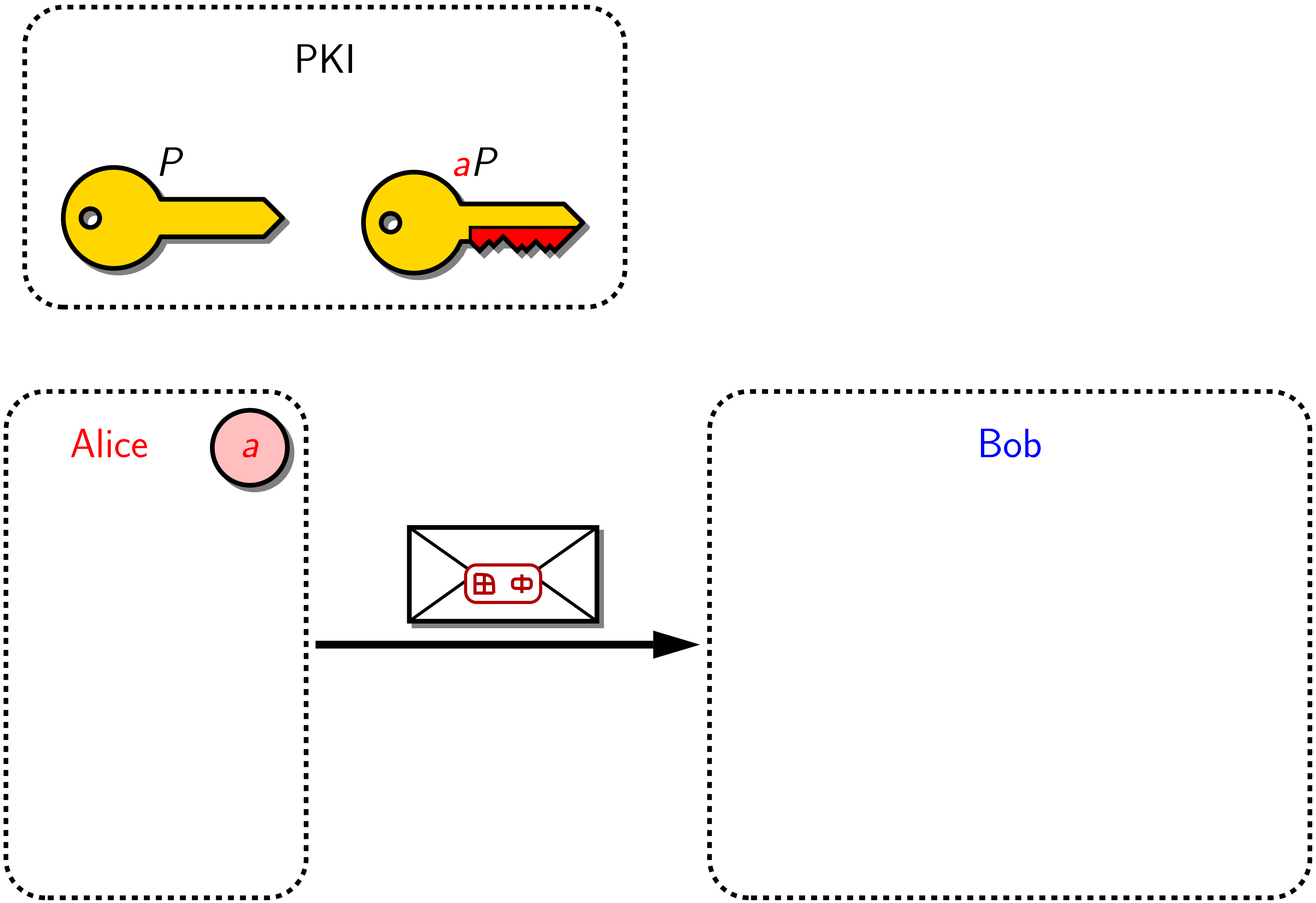
Short signature (Boneh, Lynn & Shacham, 2001)



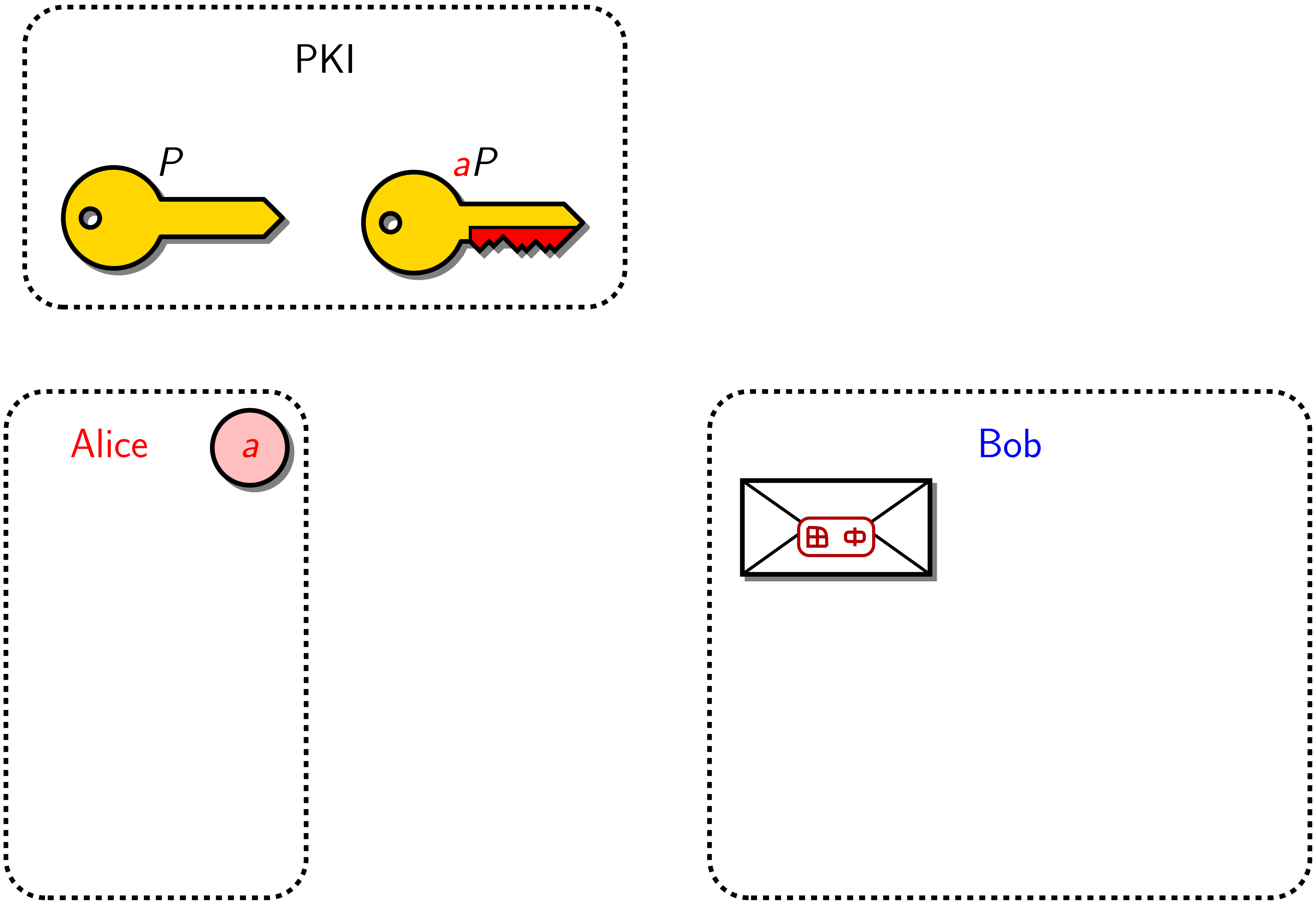
Short signature (Boneh, Lynn & Shacham, 2001)



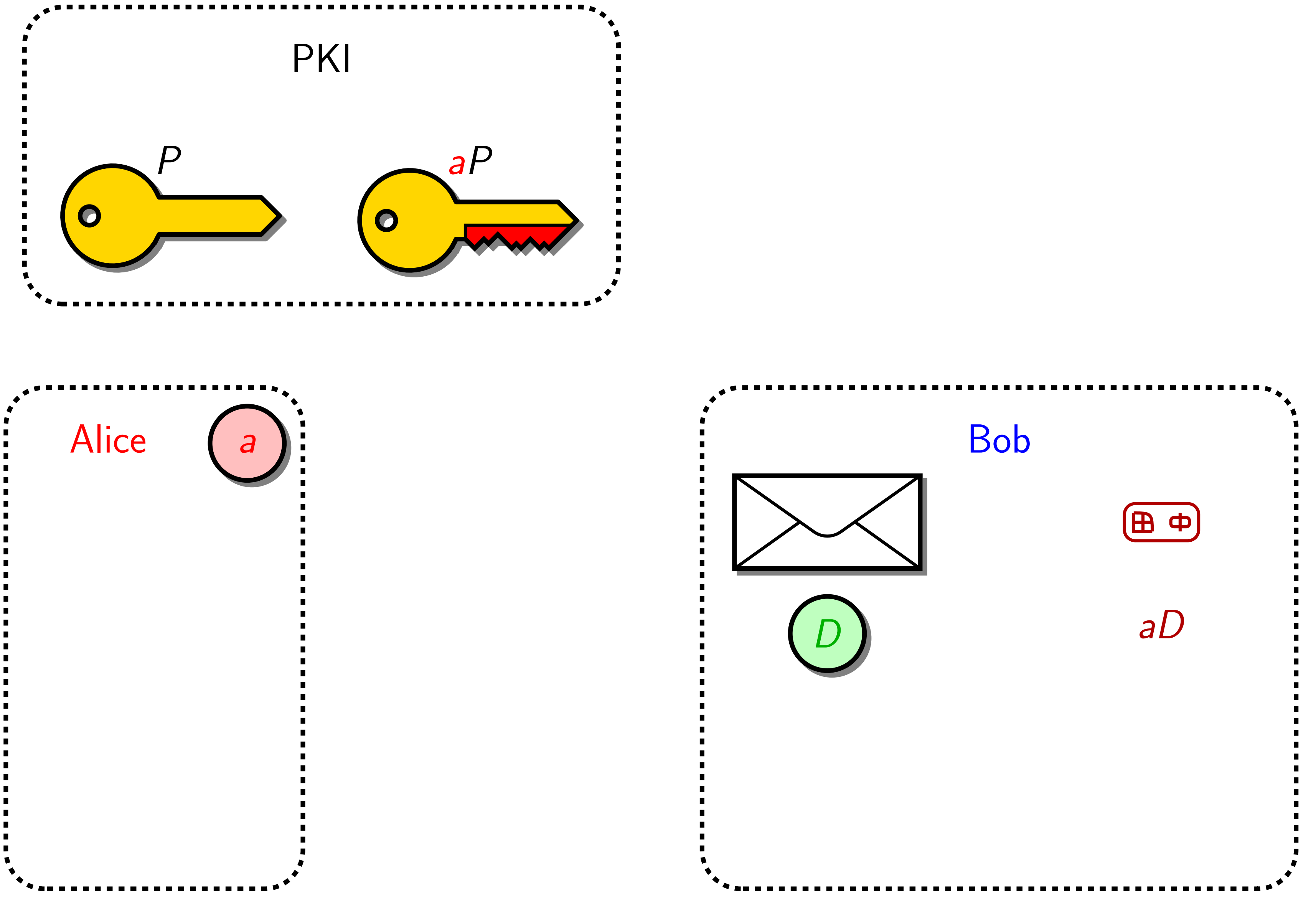
Short signature (Boneh, Lynn & Shacham, 2001)



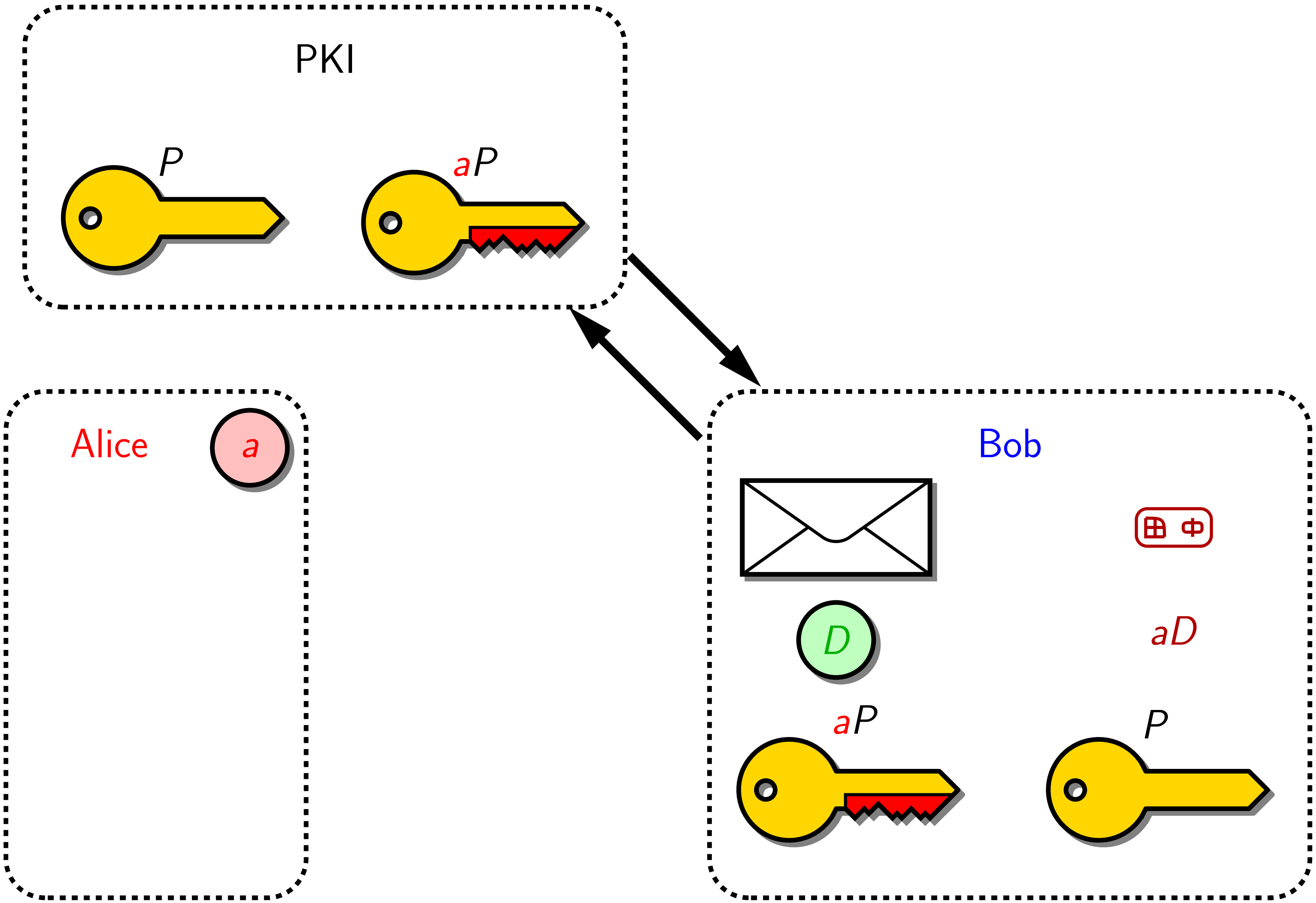
Short signature (Boneh, Lynn & Shacham, 2001)



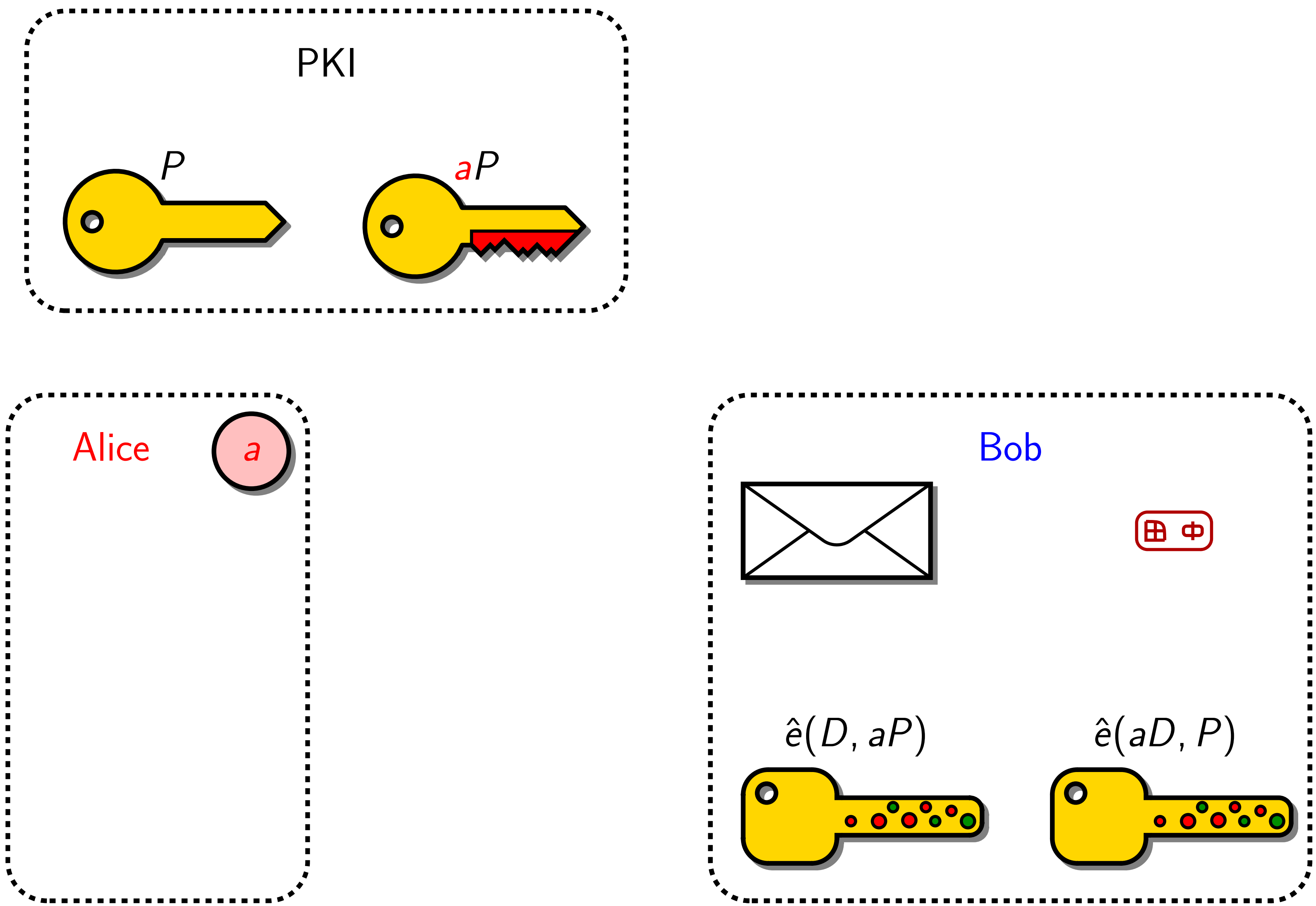
Short signature (Boneh, Lynn & Shacham, 2001)



Short signature (Boneh, Lynn & Shacham, 2001)



Short signature (Boneh, Lynn & Shacham, 2001)



Outline of the talk

- Pairing-based cryptography
- Pairings over elliptic curves
- Finite-field arithmetic
- Implementation results
- Concluding thoughts

Pairings over elliptic curves

- We first define
  - $\mathbb{F}_q$, a finite field, with $q = 2^m$, 3^m or p
  - E, an elliptic curve defined over $\mathbb{F}_q$
  - ℓ, a large prime factor of $\#E(\mathbb{F}_q)$

Pairings over elliptic curves

- We first define
  - $\mathbb{F}_q$, a finite field, with $q = 2^m$, 3^m or p
  - E, an elliptic curve defined over $\mathbb{F}_q$
  - ℓ, a large prime factor of $\#E(\mathbb{F}_q)$
- $\mathbb{G}_1 = E(\mathbb{F}_q)[\ell]$, the $\mathbb{F}_q$-rational ℓ-torsion of E:

$$\mathbb{G}_1 = \{P \in E(\mathbb{F}_q) \mid \ell P = \mathcal{O}\}$$

Pairings over elliptic curves

- We first define
  - $\mathbb{F}_q$, a finite field, with $q = 2^m$, 3^m or p
  - E, an elliptic curve defined over $\mathbb{F}_q$
  - ℓ, a large prime factor of $\#E(\mathbb{F}_q)$
- $\mathbb{G}_1 = E(\mathbb{F}_q)[\ell]$, the $\mathbb{F}_q$-rational ℓ-torsion of E:

$$\mathbb{G}_1 = \{P \in E(\mathbb{F}_q) \mid \ell P = \mathcal{O}\}$$

- $\mathbb{G}_2 = \mu_\ell$, the group of ℓ-th roots of unity in $\mathbb{F}_{q^k}^\times$:

$$\mathbb{G}_2 = \{U \in \mathbb{F}_{q^k}^\times \mid U^\ell = 1\}$$

Pairings over elliptic curves

- We first define
  - $\mathbb{F}_q$, a finite field, with $q = 2^m$, 3^m or p
  - E, an elliptic curve defined over $\mathbb{F}_q$
  - ℓ, a large prime factor of $\#E(\mathbb{F}_q)$
- $\mathbb{G}_1 = E(\mathbb{F}_q)[\ell]$, the $\mathbb{F}_q$-rational ℓ-torsion of E:

$$\mathbb{G}_1 = \{P \in E(\mathbb{F}_q) \mid \ell P = \mathcal{O}\}$$

- $\mathbb{G}_2 = \mu_\ell$, the group of ℓ-th roots of unity in $\mathbb{F}_{q^k}^\times$:

$$\mathbb{G}_2 = \{U \in \mathbb{F}_{q^k}^\times \mid U^\ell = 1\}$$

- k is the embedding degree, the smallest integer such that $\mu_\ell \subseteq \mathbb{F}_{q^k}^\times$
  - usually large for ordinary elliptic curves
  - bounded in the case of supersingular elliptic curves (4 in characteristic 2; 6 in characteristic 3; and 2 in characteristic > 3)

Pairings over elliptic curves

- We first define
  - $\mathbb{F}_q$, a finite field, with $q = 2^m$, 3^m or p
  - E, an elliptic curve defined over $\mathbb{F}_q$
  - ℓ, a large prime factor of $\#E(\mathbb{F}_q)$
- $\mathbb{G}_1 = E(\mathbb{F}_q)[\ell]$, the $\mathbb{F}_q$-rational ℓ-torsion of E:

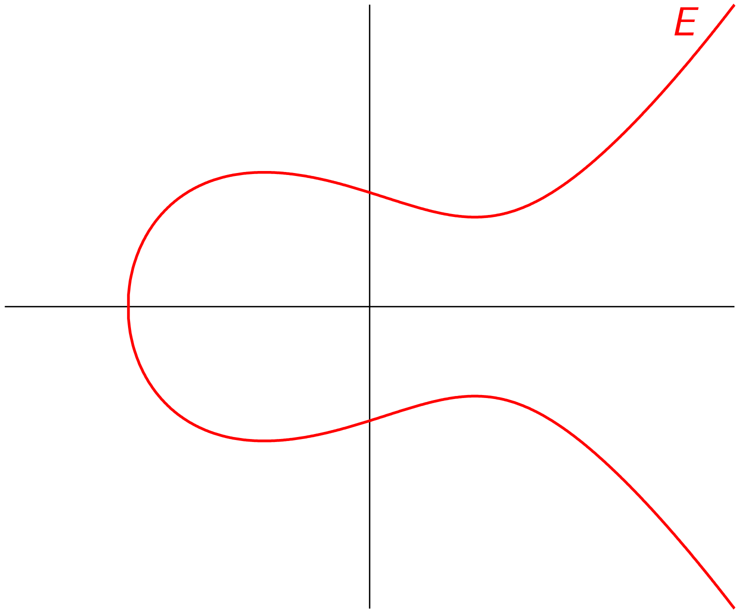
$$\mathbb{G}_1 = \{P \in E(\mathbb{F}_q) \mid \ell P = \mathcal{O}\}$$

- $\mathbb{G}_2 = \mu_\ell$, the group of ℓ-th roots of unity in $\mathbb{F}_{q^k}^\times$:

$$\mathbb{G}_2 = \{U \in \mathbb{F}_{q^k}^\times \mid U^\ell = 1\}$$

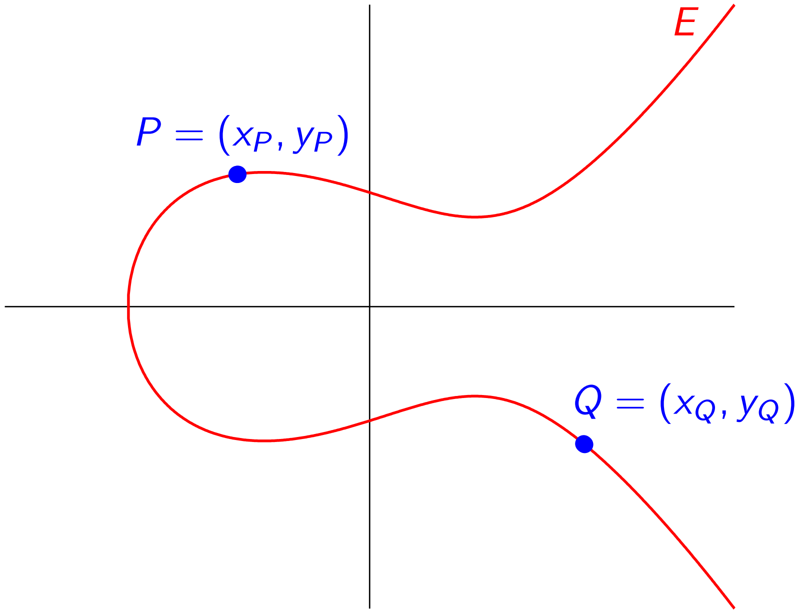
- k is the embedding degree, the smallest integer such that $\mu_\ell \subseteq \mathbb{F}_{q^k}^\times$
  - usually large for ordinary elliptic curves
  - bounded in the case of supersingular elliptic curves
    (4 in characteristic 2; 6 in characteristic 3; and 2 in characteristic > 3)

The Tate pairing



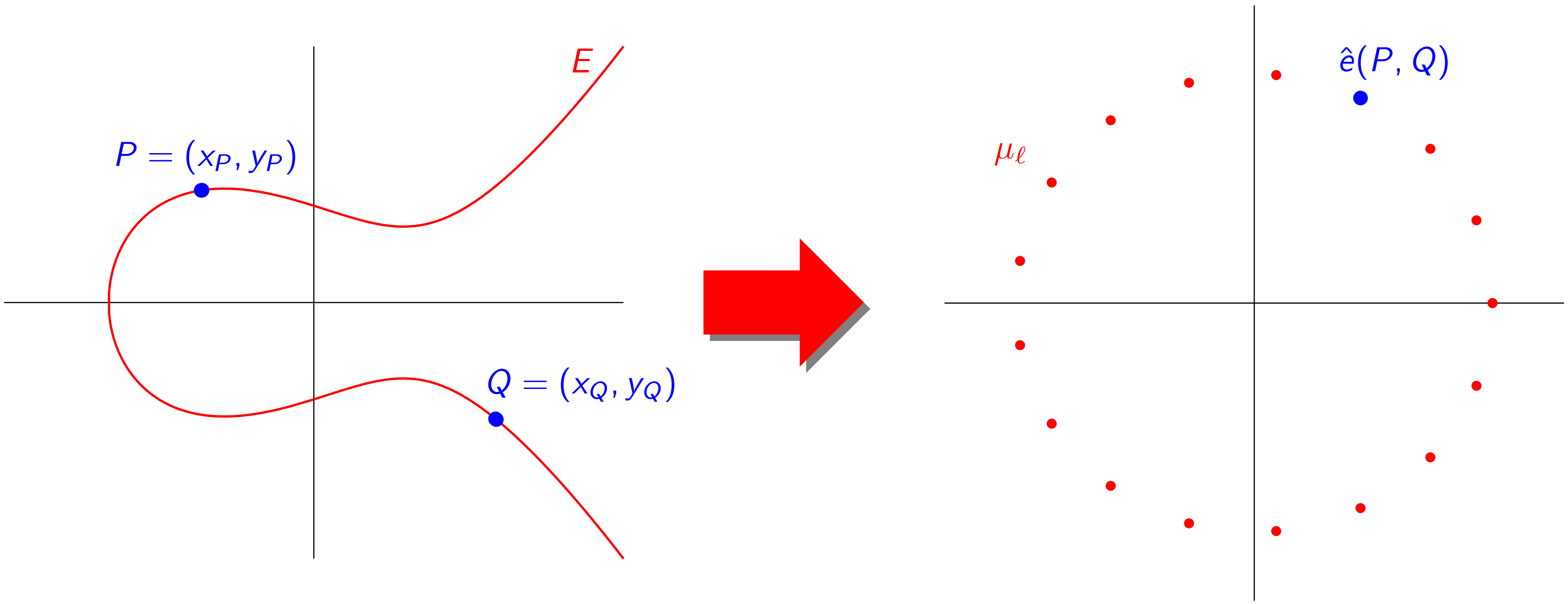
$\hat{e}$

The Tate pairing



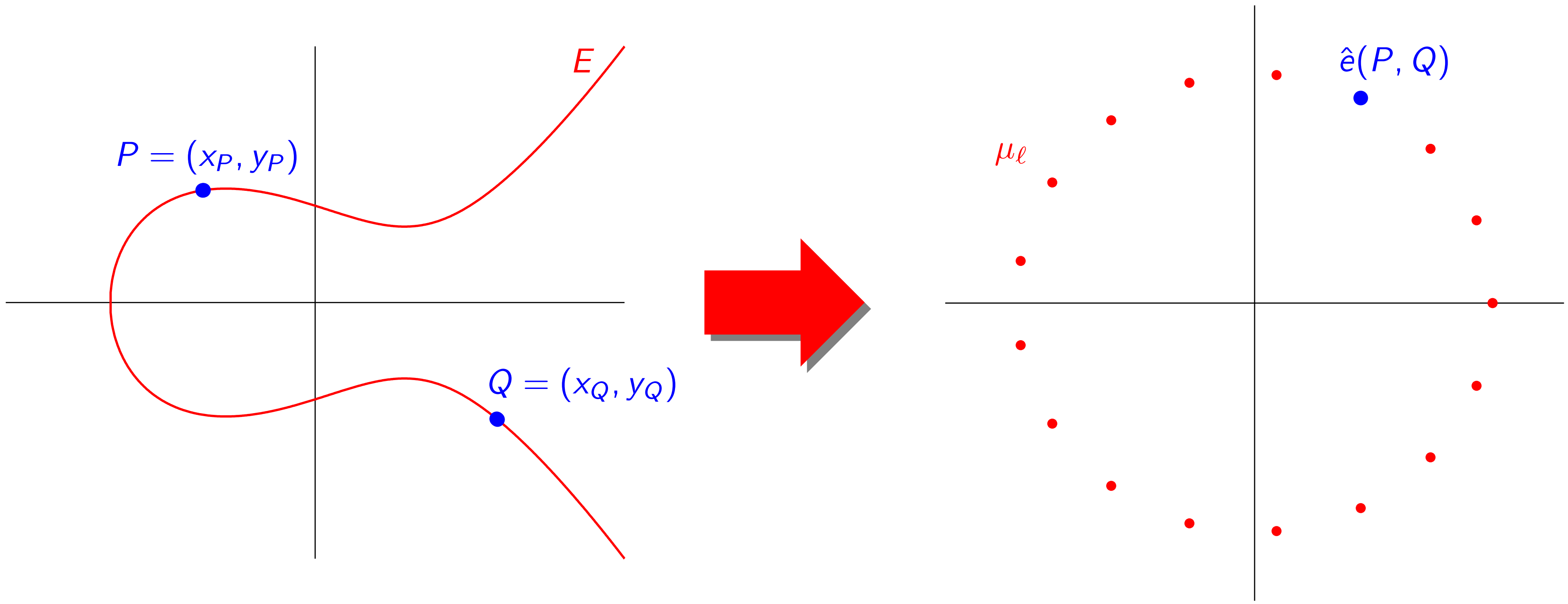
$$\begin{array}{ccccc} \hat{e} & : & E(\mathbb{F}_q)[\ell] & \times & E(\mathbb{F}_q)[\ell] \\ & & (\quad P & , & Q \quad) \end{array}$$

The Tate pairing



$$
\begin{array}{rccccc}
\hat{e} & : & E(\mathbb{F}_q)[\ell] \times E(\mathbb{F}_q)[\ell] & \longrightarrow & \mu_\ell & \subseteq \mathbb{F}_{q^k}^{\times} \\
 & & (\quad P \quad , \quad Q \quad) & \longmapsto & \hat{e}(P, Q) &
\end{array}
$$

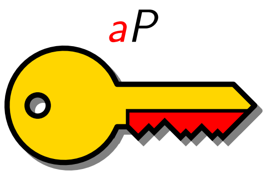
The Tate pairing



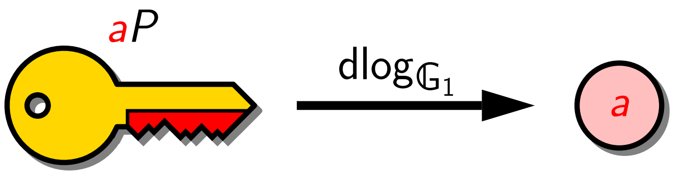
$$\begin{array}{rccccc} \hat{e} & : & E(\mathbb{F}_q)[\ell] \times E(\mathbb{F}_q)[\ell] & \longrightarrow & \mu_\ell & \subseteq \mathbb{F}_{q^k}^{\times} \\ & & (\ P\ ,\ Q\ ) & \longmapsto & \hat{e}(P, Q) & \end{array}$$

- Computation via Miller's iterative algorithm:
  - $m/2$ iterations over $\mathbb{F}_{2^m}$ and $\mathbb{F}_{3^m}$ (η_T pairing)
  - $\log_2 p$ iterations over $\mathbb{F}_p$

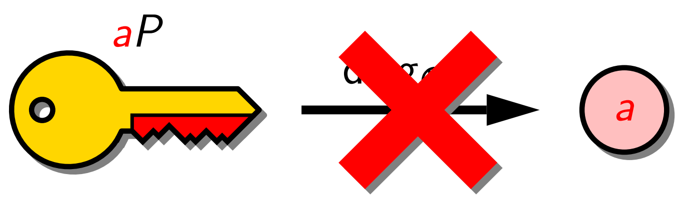
Security considerations



Security considerations

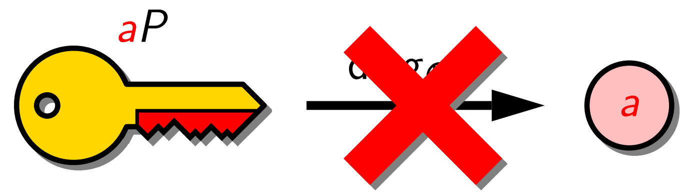


Security considerations

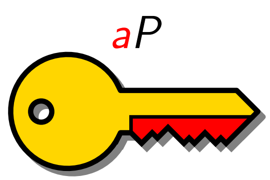


- Discrete logarithm problem should be hard in $\mathbb{G}_1$

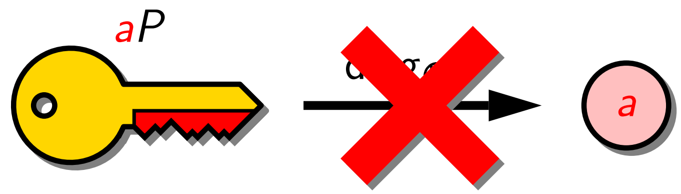
Security considerations



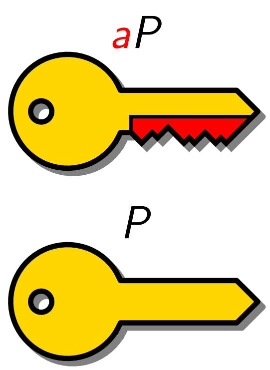
- Discrete logarithm problem should be hard in $\mathbb{G}_1$



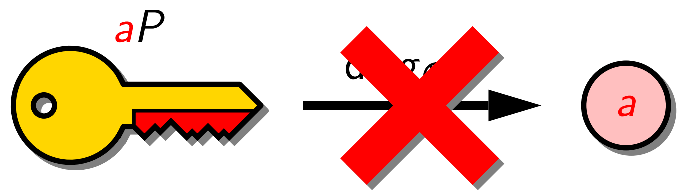
Security considerations



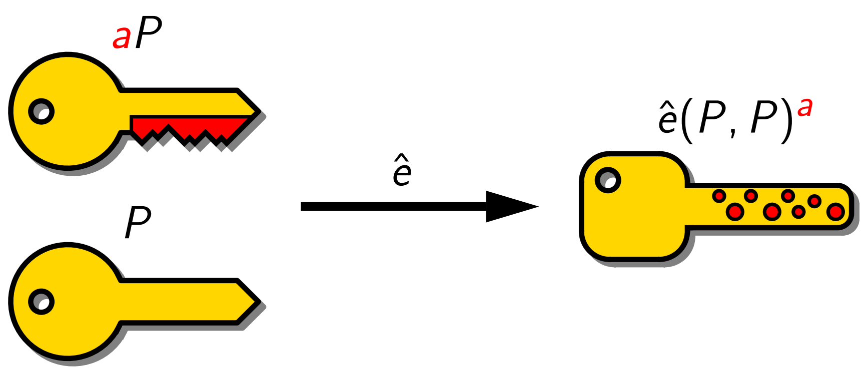
- Discrete logarithm problem should be hard in $\mathbb{G}_1$



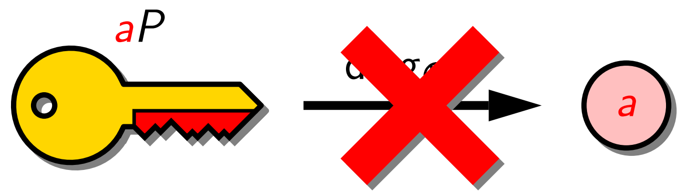
Security considerations



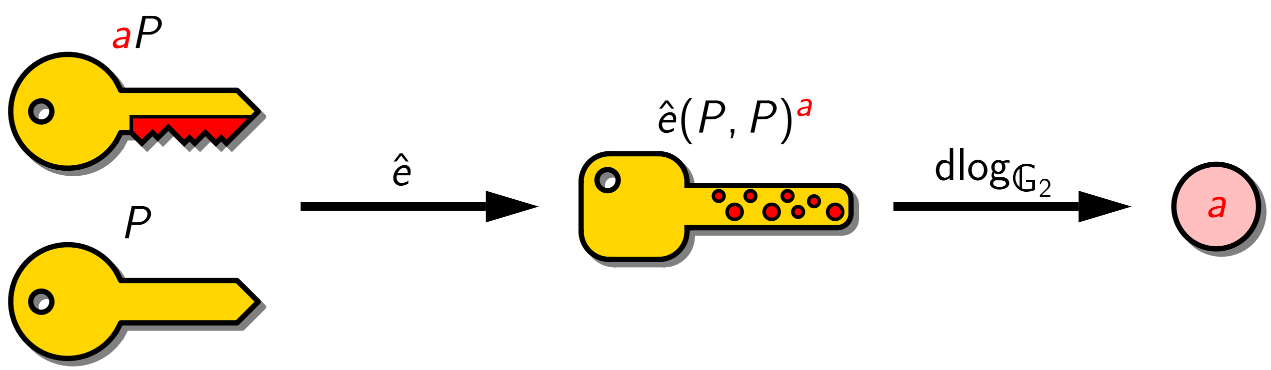
- Discrete logarithm problem should be hard in $\mathbb{G}_1$



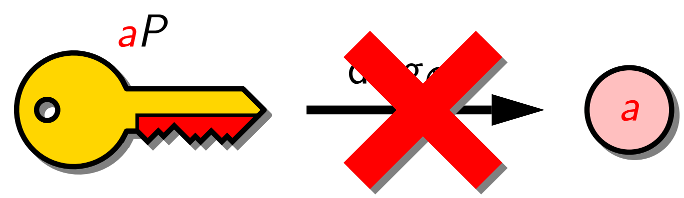
Security considerations



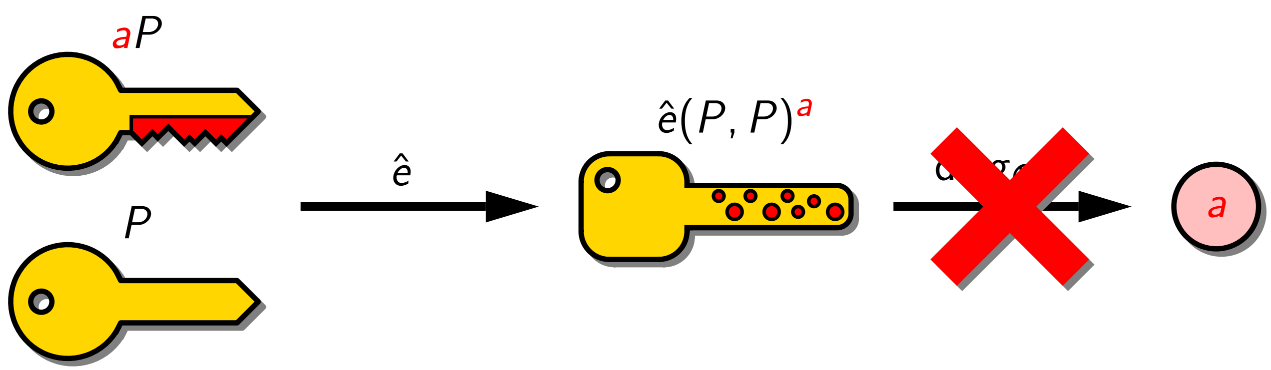
- Discrete logarithm problem should be hard in $\mathbb{G}_1$



Security considerations



- Discrete logarithm problem should be hard in $\mathbb{G}_1$



- Discrete logarithm problem should be hard in $\mathbb{G}_2$

Security considerations

$$\hat{e} : E(\mathbb{F}_q)[\ell] \times E(\mathbb{F}_q)[\ell] \longrightarrow \mu_\ell \subseteq \mathbb{F}_{q^k}^\times$$

Security considerations

$$\hat{e} : E(\mathbb{F}_q)[\ell] \times E(\mathbb{F}_q)[\ell] \rightarrow \mu_\ell \subseteq \mathbb{F}_{q^k}^\times$$

- Discrete logarithm in $\mathbb{G}_1 = E(\mathbb{F}_q)[\ell]$ (Pollard's ρ):

$$\sqrt{\ell} \approx \sqrt{q}$$

- Discrete logarithm in $\mathbb{G}_2 = \mu_\ell \subseteq \mathbb{F}_{q^k}^\times$ (FFS or NFS):

$$\exp\left(c \cdot (\ln q^k)^{\frac{1}{3}} \cdot (\ln \ln q^k)^{\frac{2}{3}}\right)$$

Security considerations

$$\hat{e} : E(\mathbb{F}_q)[\ell] \times E(\mathbb{F}_q)[\ell] \longrightarrow \mu_\ell \subseteq \mathbb{F}_{q^k}^\times$$

- Discrete logarithm in $\mathbb{G}_1 = E(\mathbb{F}_q)[\ell]$ (Pollard's ρ):

$$\sqrt{\ell} \approx \sqrt{q} = \exp\left(\frac{1}{2} \cdot (\ln q)\right)$$

- Discrete logarithm in $\mathbb{G}_2 = \mu_\ell \subseteq \mathbb{F}_{q^k}^\times$ (FFS or NFS):

$$\exp\left(c \cdot (\ln q^k)^{\frac{1}{3}} \cdot (\ln \ln q^k)^{\frac{2}{3}}\right)$$

Security considerations

$$\hat{e} : E(\mathbb{F}_q)[\ell] \times E(\mathbb{F}_q)[\ell] \rightarrow \mu_\ell \subseteq \mathbb{F}_{q^k}^\times$$

- Discrete logarithm in $\mathbb{G}_1 = E(\mathbb{F}_q)[\ell]$ (Pollard's ρ):

$$\sqrt{\ell} \approx \sqrt{q} = \exp\left(\frac{1}{2} \cdot (\ln q)\right)$$

- Discrete logarithm in $\mathbb{G}_2 = \mu_\ell \subseteq \mathbb{F}_{q^k}^\times$ (FFS or NFS):

$$\exp\left(c \cdot (\ln q^k)^{\frac{1}{3}} \cdot (\ln \ln q^k)^{\frac{2}{3}}\right)$$

- The discrete logarithm problem is usually easier in $\mathbb{G}_2$ than in $\mathbb{G}_1$
  - current security: $\sim 2^{80}$, equivalent to 80-bit symmetric encryption or RSA-1024
  - recommended security: $\sim 2^{128}$ (AES-128, RSA-3072)

Security considerations

$$\hat{e} : E(\mathbb{F}_q)[\ell] \times E(\mathbb{F}_q)[\ell] \longrightarrow \mu_\ell \subseteq \mathbb{F}_{q^k}^\times$$

- The embedding degree k depends on the field characteristic q

Security considerations

$$\hat{e} : E(\mathbb{F}_q)[\ell] \times E(\mathbb{F}_q)[\ell] \rightarrow \mu_\ell \subseteq \mathbb{F}_{q^k}^\times$$

- The embedding degree k depends on the field characteristic q

Base field $(\mathbb{F}_q)$	$\mathbb{F}_{2^m}$	$\mathbb{F}_{3^m}$	$\mathbb{F}_p$
Embedding degree (k)	4	6	2

Security considerations

$$\hat{e} : E(\mathbb{F}_q)[\ell] \times E(\mathbb{F}_q)[\ell] \rightarrow \mu_\ell \subseteq \mathbb{F}_{q^k}^\times$$

- The embedding degree k depends on the field characteristic q

Base field ($\mathbb{F}_q$)	$\mathbb{F}_{2^m}$	$\mathbb{F}_{3^m}$	$\mathbb{F}_p$
Embedding degree (k)	4	6	2
Lower security ($\sim 2^{64}$)	$m = 239$	$m = 97$	$\|p\| \approx 256$ bits
Medium security ($\sim 2^{80}$)	$m = 373$	$m = 163$	$\|p\| \approx 512$ bits
Higher security ($\sim 2^{128}$)	$m = 1103$	$m = 503$	$\|p\| \approx 1536$ bits

Security considerations

$$\hat{e} : E(\mathbb{F}_q)[\ell] \times E(\mathbb{F}_q)[\ell] \rightarrow \mu_\ell \subseteq \mathbb{F}_{q^k}^\times$$

- The embedding degree k depends on the field characteristic q

Base field ($\mathbb{F}_q$)	$\mathbb{F}_{2^m}$	$\mathbb{F}_{3^m}$	$\mathbb{F}_p$
Embedding degree (k)	4	6	2
Lower security ($\sim 2^{64}$)	$m = 239$	$m = 97$	$\lvert p \rvert \approx 256$ bits
Medium security ($\sim 2^{80}$)	$m = 373$	$m = 163$	$\lvert p \rvert \approx 512$ bits
Higher security ($\sim 2^{128}$)	$m = 1103$	$m = 503$	$\lvert p \rvert \approx 1536$ bits

- $\mathbb{F}_{2^m}$: simpler finite field arithmetic

Security considerations

$$\hat{e} : E(\mathbb{F}_q)[\ell] \times E(\mathbb{F}_q)[\ell] \rightarrow \mu_\ell \subseteq \mathbb{F}_{q^k}^\times$$

- The embedding degree k depends on the field characteristic q

Base field ($\mathbb{F}_q$)	$\mathbb{F}_{2^m}$	$\mathbb{F}_{3^m}$	$\mathbb{F}_p$
Embedding degree (k)	4	6	2
Lower security ($\sim 2^{64}$)	$m = 239$	$m = 97$	$\lvert p \rvert \approx 256$ bits
Medium security ($\sim 2^{80}$)	$m = 373$	$m = 163$	$\lvert p \rvert \approx 512$ bits
Higher security ($\sim 2^{128}$)	$m = 1103$	$m = 503$	$\lvert p \rvert \approx 1536$ bits

- $\mathbb{F}_{2^m}$: simpler finite field arithmetic
- $\mathbb{F}_{3^m}$: smaller field extension

Security considerations

$$\hat{e} : E(\mathbb{F}_q)[\ell] \times E(\mathbb{F}_q)[\ell] \rightarrow \mu_\ell \subseteq \mathbb{F}_{q^k}^\times$$

- The embedding degree k depends on the field characteristic q

Base field $(\mathbb{F}_q)$	$\mathbb{F}_{2^m}$	$\mathbb{F}_{3^m}$	$\mathbb{F}_p$
Embedding degree (k)	4	6	2
Lower security $(\sim 2^{64})$	$m = 239$	$m = 97$	$\lvert p \rvert \approx 256$ bits
Medium security $(\sim 2^{80})$	$m = 373$	$m = 163$	$\lvert p \rvert \approx 512$ bits
Higher security $(\sim 2^{128})$	$m = 1103$	$m = 503$	$\lvert p \rvert \approx 1536$ bits

- $\mathbb{F}_{2^m}$: simpler finite field arithmetic
- $\mathbb{F}_{3^m}$: smaller field extension
- $\mathbb{F}_p$: prohibitive field sizes

Security considerations

$$\hat{e} : E(\mathbb{F}_q)[\ell] \times E(\mathbb{F}_q)[\ell] \rightarrow \mu_\ell \subseteq \mathbb{F}_{q^k}^\times$$

- The embedding degree k depends on the field characteristic q

Base field ($\mathbb{F}_q$)	$\mathbb{F}_{2^m}$	$\mathbb{F}_{3^m}$	$\mathbb{F}_p$
Embedding degree (k)	4	6	2
Lower security ($\sim 2^{64}$)	$m = 239$	$m = 97$	$\lvert p \rvert \approx 256$ bits
Medium security ($\sim 2^{80}$)	$m = 373$	$m = 163$	$\lvert p \rvert \approx 512$ bits
Higher security ($\sim 2^{128}$)	$m = 1103$	$m = 503$	$\lvert p \rvert \approx 1536$ bits

- $\mathbb{F}_{2^m}$: simpler finite field arithmetic
- $\mathbb{F}_{3^m}$: smaller field extension
- $\mathbb{F}_p$: prohibitive field sizes

Computation of the Tate pairing

$$\hat{e} : E(\mathbb{F}_{p^m})[\ell] \times E(\mathbb{F}_{p^m})[\ell] \longrightarrow \mu_\ell \subseteq \mathbb{F}_{p^{km}}^{\times}$$

Computation of the Tate pairing

$$\hat{e} : E(\mathbb{F}_{p^m})[\ell] \times E(\mathbb{F}_{p^m})[\ell] \rightarrow \mu_\ell \subseteq \mathbb{F}_{p^{km}}^\times$$

- Arithmetic over $\mathbb{F}_{p^m}$:
  - polynomial basis: $\mathbb{F}_{p^m} \cong \mathbb{F}_p[x]/(f(x))$
  - $f(x)$, degree-m polynomial irreducible over $\mathbb{F}_p$

Computation of the Tate pairing

$$\hat{e} : E(\mathbb{F}_{p^m})[\ell] \times E(\mathbb{F}_{p^m})[\ell] \rightarrow \mu_\ell \subseteq \mathbb{F}_{p^{km}}^{\times}$$

- Arithmetic over $\mathbb{F}_{p^m}$:
  - polynomial basis: $\mathbb{F}_{p^m} \cong \mathbb{F}_p[x]/(f(x))$
  - $f(x)$, degree-m polynomial irreducible over $\mathbb{F}_p$
- Arithmetic over $\mathbb{F}_{p^{km}}^{\times}$:
  - tower-field representation
  - only arithmetic over the underlying field $\mathbb{F}_{p^m}$

Computation of the Tate pairing

$$\hat{e} : E(\mathbb{F}_{p^m})[\ell] \times E(\mathbb{F}_{p^m})[\ell] \rightarrow \mu_\ell \subseteq \mathbb{F}_{p^{km}}^\times$$

- Arithmetic over $\mathbb{F}_{p^m}$:
  - polynomial basis: $\mathbb{F}_{p^m} \cong \mathbb{F}_p[x]/(f(x))$
  - $f(x)$, degree-m polynomial irreducible over $\mathbb{F}_p$
- Arithmetic over $\mathbb{F}_{p^{km}}^\times$:
  - tower-field representation
  - only arithmetic over the underlying field $\mathbb{F}_{p^m}$
- Operations over $\mathbb{F}_{p^m}$:

	Characteristic 2		**Characteristic 3**	
Base field ($\mathbb{F}_{p^m}$)	$\mathbb{F}_{2^m}$	$\mathbb{F}_{2^{313}}$	$\mathbb{F}_{3^m}$	$\mathbb{F}_{3^{127}}$
$+/-$	$27\lfloor \frac{m}{2} \rfloor + 75$	4287	$119\lfloor \frac{m}{4} \rfloor + 260$	3949
$\times$	$7\lfloor \frac{m}{2} \rfloor + 29$	1121	$25\lfloor \frac{m}{4} \rfloor + 93$	868
a^p	$6m + 9$	1887	$17\lfloor \frac{m}{2} \rfloor + 8$	1079
a^{-1}	1	1	1	1

Computation of the Tate pairing

$$\hat{e} : E(\mathbb{F}_{p^m})[\ell] \times E(\mathbb{F}_{p^m})[\ell] \rightarrow \mu_\ell \subseteq \mathbb{F}_{p^{km}}^\times$$

- Arithmetic over $\mathbb{F}_{p^m}$:
  - polynomial basis: $\mathbb{F}_{p^m} \cong \mathbb{F}_p[x]/(f(x))$
  - $f(x)$, degree-m polynomial irreducible over $\mathbb{F}_p$
- Arithmetic over $\mathbb{F}_{p^{km}}^\times$:
  - tower-field representation
  - only arithmetic over the underlying field $\mathbb{F}_{p^m}$
- Operations over $\mathbb{F}_{p^m}$:

	Characteristic 2		Characteristic 3	
Base field ($\mathbb{F}_{p^m}$)	$\mathbb{F}_{2^m}$	$\mathbb{F}_{2^{313}}$	$\mathbb{F}_{3^m}$	$\mathbb{F}_{3^{127}}$
$+/-$	$27\lfloor \frac{m}{2} \rfloor + 75$	4287	$119\lfloor \frac{m}{4} \rfloor + 260$	3949
$\times$	$7\lfloor \frac{m}{2} \rfloor + 29$	1121	$25\lfloor \frac{m}{4} \rfloor + 93$	868
a^p	$6m + 9$	1887	$17\lfloor \frac{m}{2} \rfloor + 8$	1079
a^{-1}	1	1	1	1

- Software not well suited to small characteristic: need for hardware acceleration

Outline of the talk

- Pairing-based cryptography
- Pairings over elliptic curves
- Finite-field arithmetic
- Implementation results
- Concluding thoughts

Outline of the talk

- Pairing-based cryptography
- Pairings over elliptic curves
- Finite-field arithmetic (only in characteristic 3)
- Implementation results
- Concluding thoughts

Arithmetic over $\mathbb{F}_{3^m}$

- $f \in \mathbb{F}_3[x]$: degree-m irreducible polynomial over $\mathbb{F}_3$

$$f = x^m + f_{m-1}x^{m-1} + \cdots + f_1 x + f_0$$

Arithmetic over $\mathbb{F}_{3^m}$

- $f \in \mathbb{F}_3[x]$: degree-m irreducible polynomial over $\mathbb{F}_3$

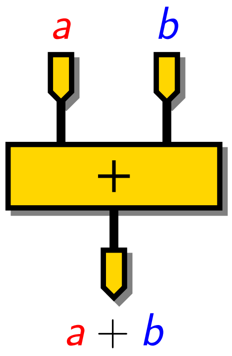
$$f = x^m + f_{m-1}x^{m-1} + \cdots + f_1 x + f_0$$

- $\mathbb{F}_{3^m} \cong \mathbb{F}_3[x]/(f)$
- $a \in \mathbb{F}_{3^m}$:

$$a = a_{m-1}x^{m-1} + \cdots + a_1 x + a_0$$

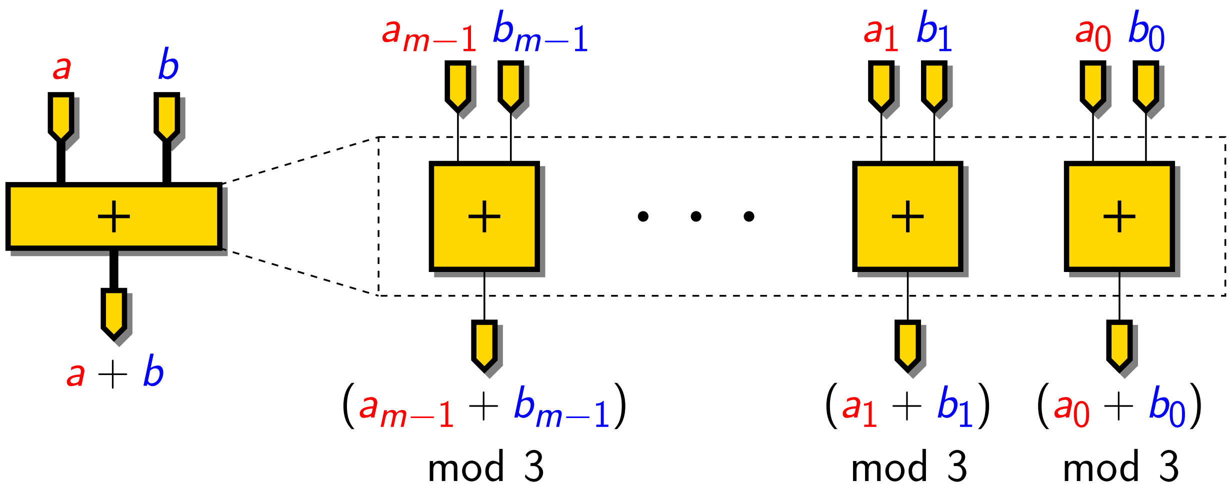
- Each element of $\mathbb{F}_3$ stored using two bits

Addition over $\mathbb{F}_{3^m}$



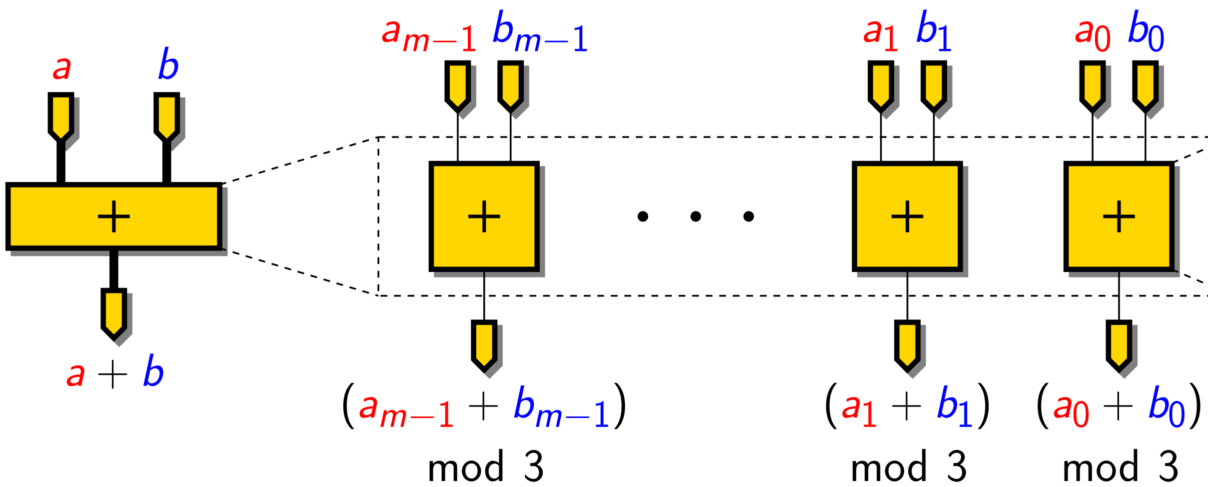
- $r = a + b = (a_{m-1} + b_{m-1})x^{m-1} + \cdots + (a_1 + b_1)x + (a_0 + b_0)$

Addition over $\mathbb{F}_{3^m}$



- $r = a + b = (a_{m-1} + b_{m-1})x^{m-1} + \cdots + (a_1 + b_1)x + (a_0 + b_0)$
  - coefficient-wise additions over $\mathbb{F}_3$: $r_i = (a_i + b_i) \bmod 3$

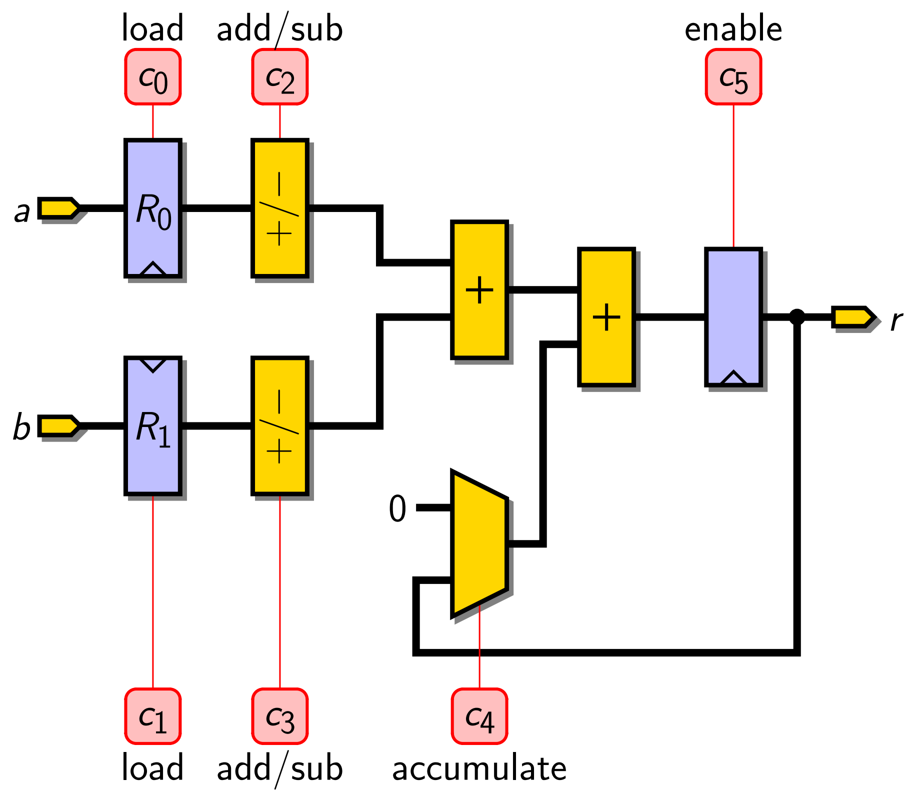
Addition over $\mathbb{F}_{3^m}$



+	0	1	2
0	0	1	2
1	1	2	0
2	2	0	1

- $r = a + b = (a_{m-1} + b_{m-1})x^{m-1} + \cdots + (a_1 + b_1)x + (a_0 + b_0)$
  - coefficient-wise additions over $\mathbb{F}_3$: $r_i = (a_i + b_i) \bmod 3$
  - addition over $\mathbb{F}_3$: small look-up tables

Addition, subtraction and accumulation over $\mathbb{F}_{3^m}$



- sign selection: multiplication by 1 or 2

$$-a \equiv 2a \pmod 3$$

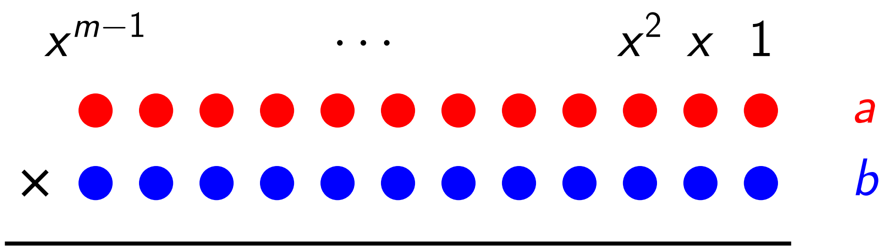
- feedback loop for accumulation

Multiplication over $\mathbb{F}_{3^m}$

- Parallel-serial multiplication
  - multiplicand loaded in a parallel register
  - multiplier loaded in a shift register
- Most significant coefficients first (Horner scheme)
- D coefficients processed at each clock cycle: $\left\lceil \frac{m}{D} \right\rceil$ cycles per multiplication

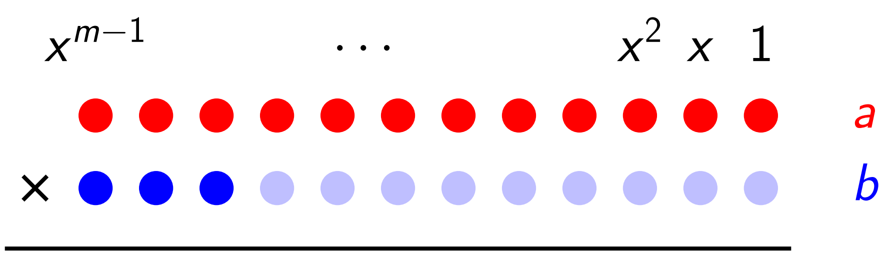
Multiplication over $\mathbb{F}_{3^m}$

- Example for $D = 3$ (3 coefficients per iteration):



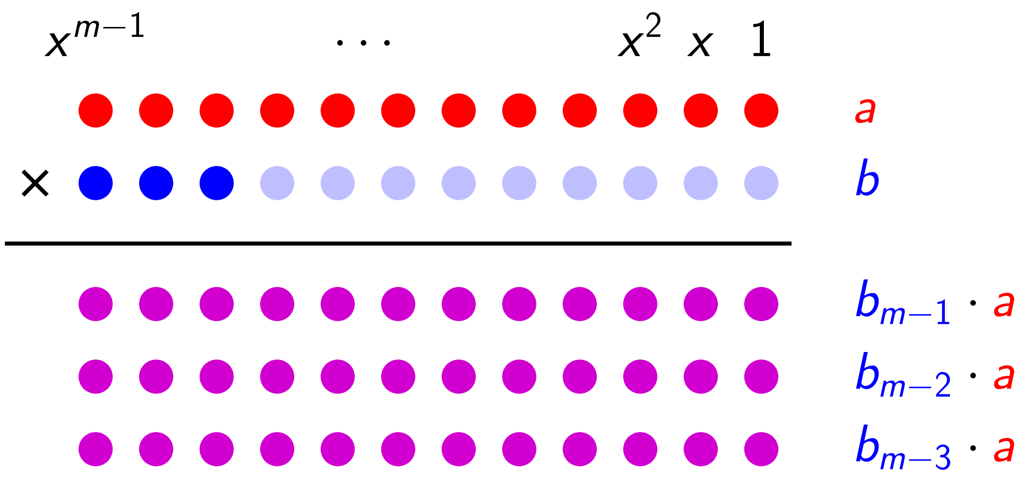
Multiplication over $\mathbb{F}_{3^m}$

- Example for $D = 3$ (3 coefficients per iteration):



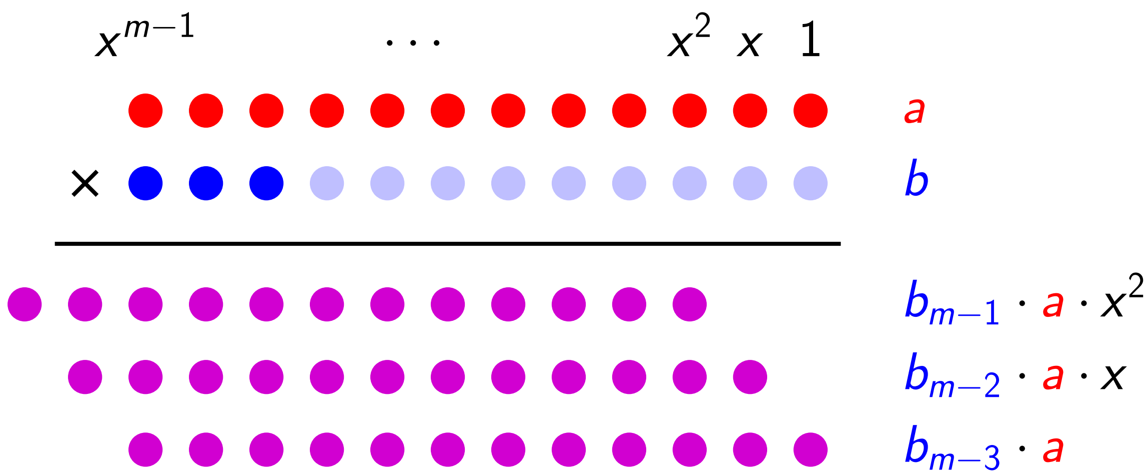
Multiplication over $\mathbb{F}_{3^m}$

- Example for $D = 3$ (3 coefficients per iteration):



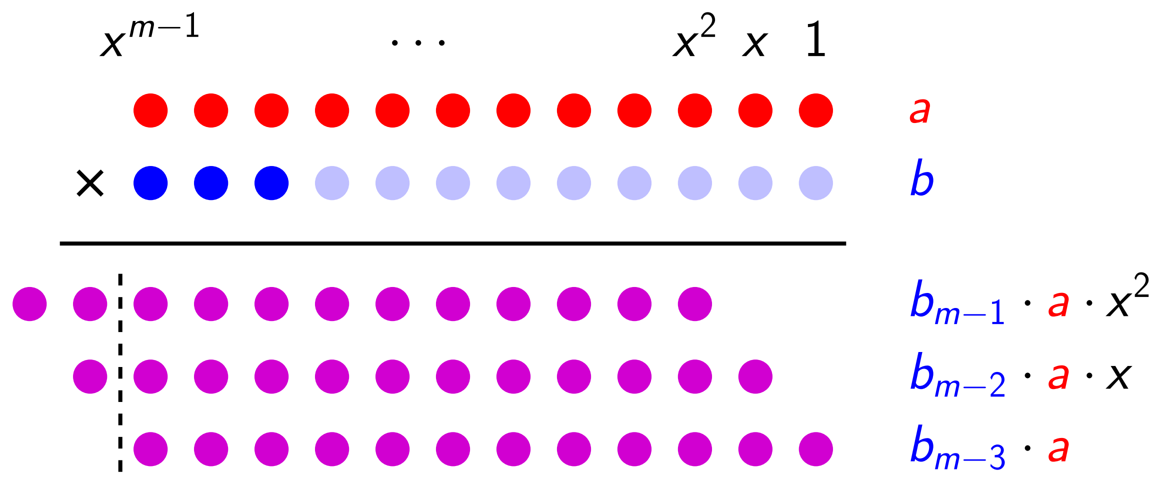
Multiplication over $\mathbb{F}_{3^m}$

- Example for $D = 3$ (3 coefficients per iteration):



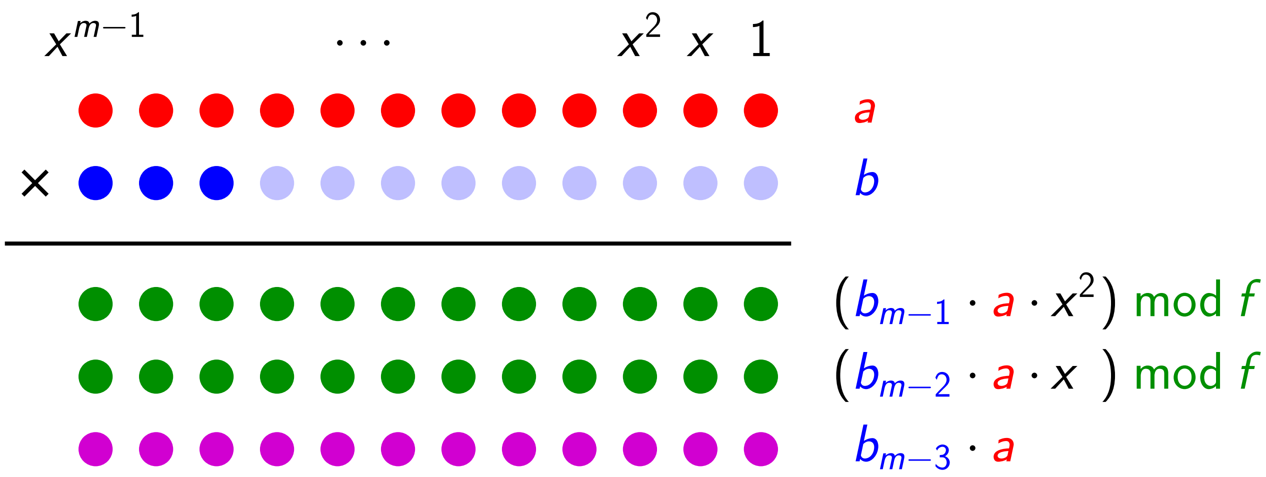
Multiplication over $\mathbb{F}_{3^m}$

- Example for $D = 3$ (3 coefficients per iteration):



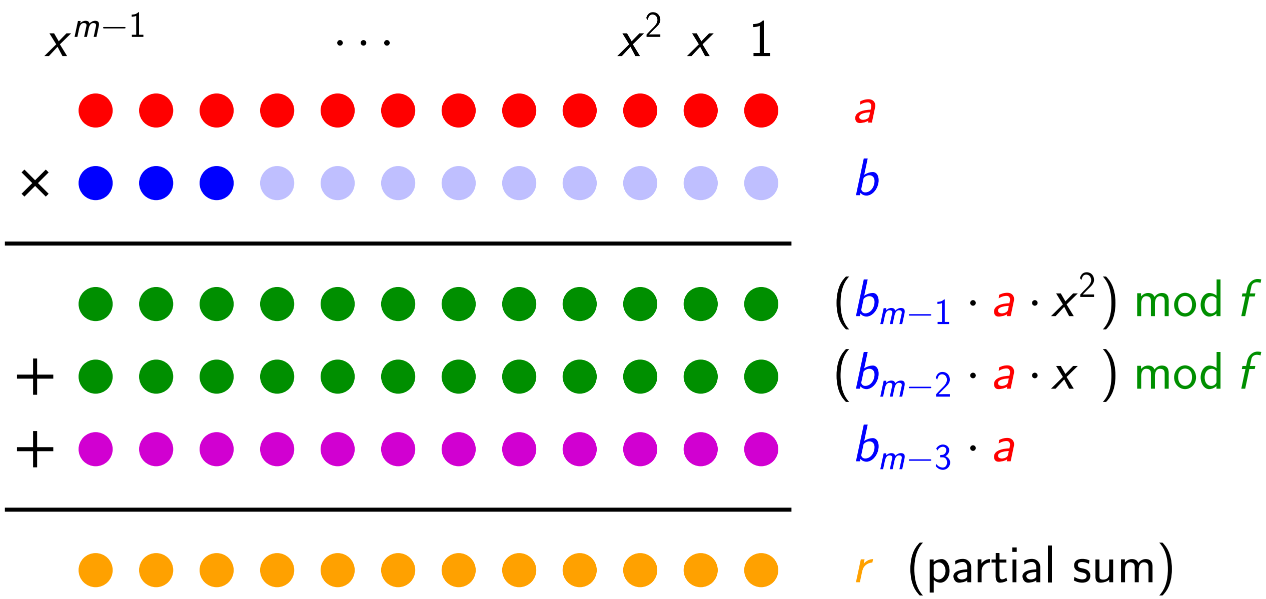
Multiplication over $\mathbb{F}_{3^m}$

- Example for $D = 3$ (3 coefficients per iteration):



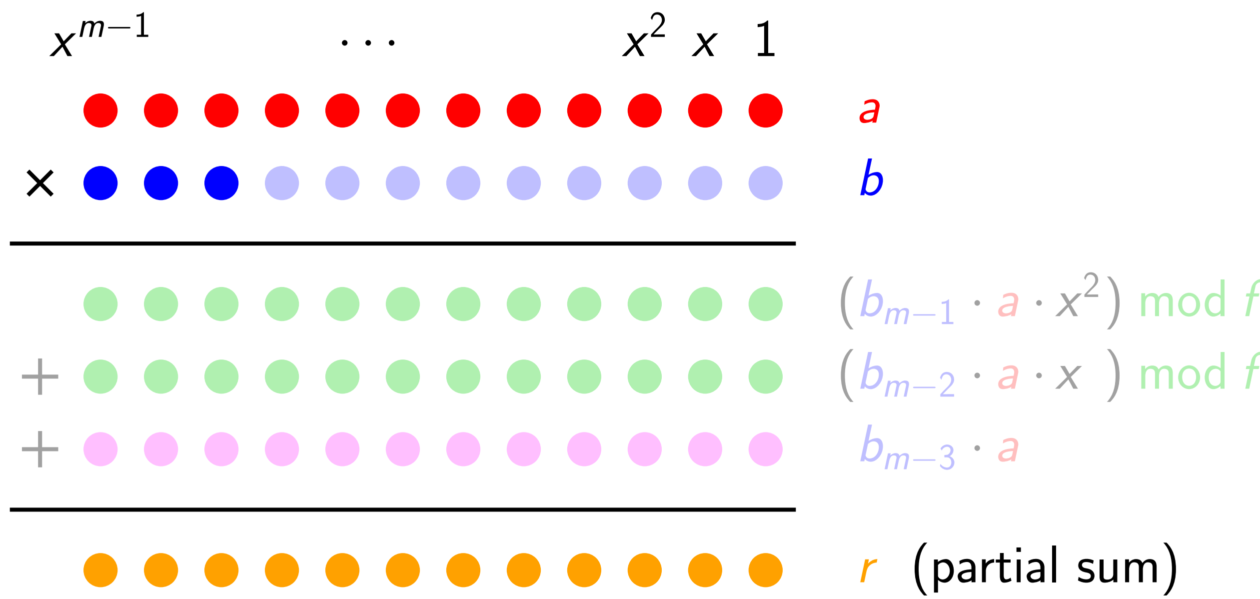
Multiplication over $\mathbb{F}_{3^m}$

- Example for $D = 3$ (3 coefficients per iteration):



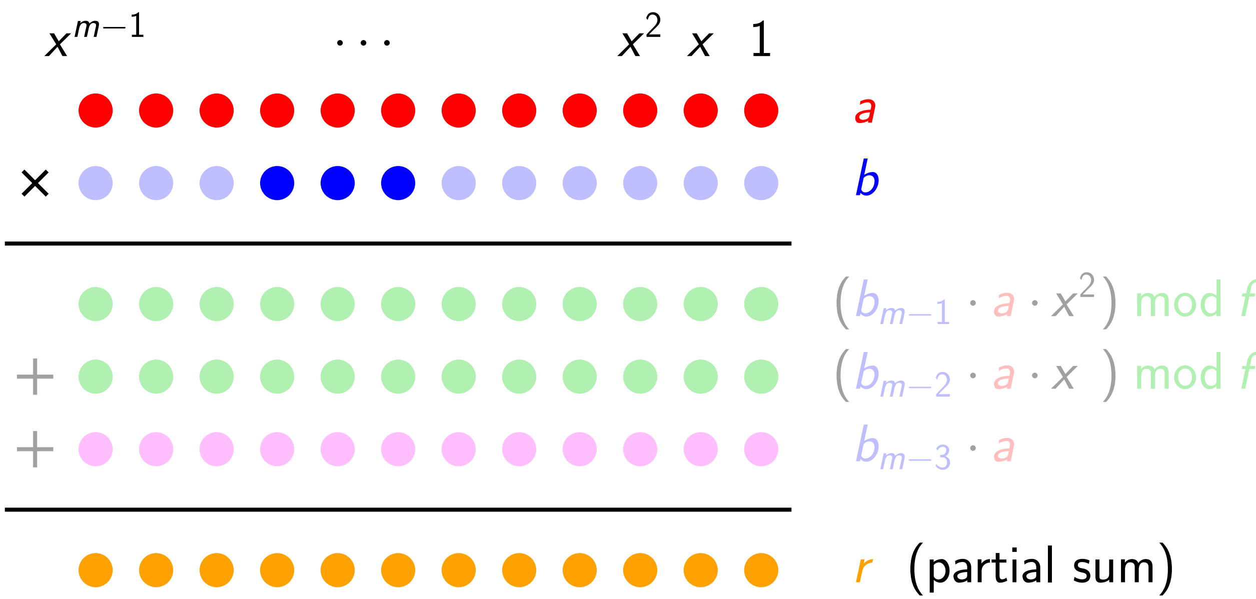
Multiplication over $\mathbb{F}_{3^m}$

- Example for $D = 3$ (3 coefficients per iteration):



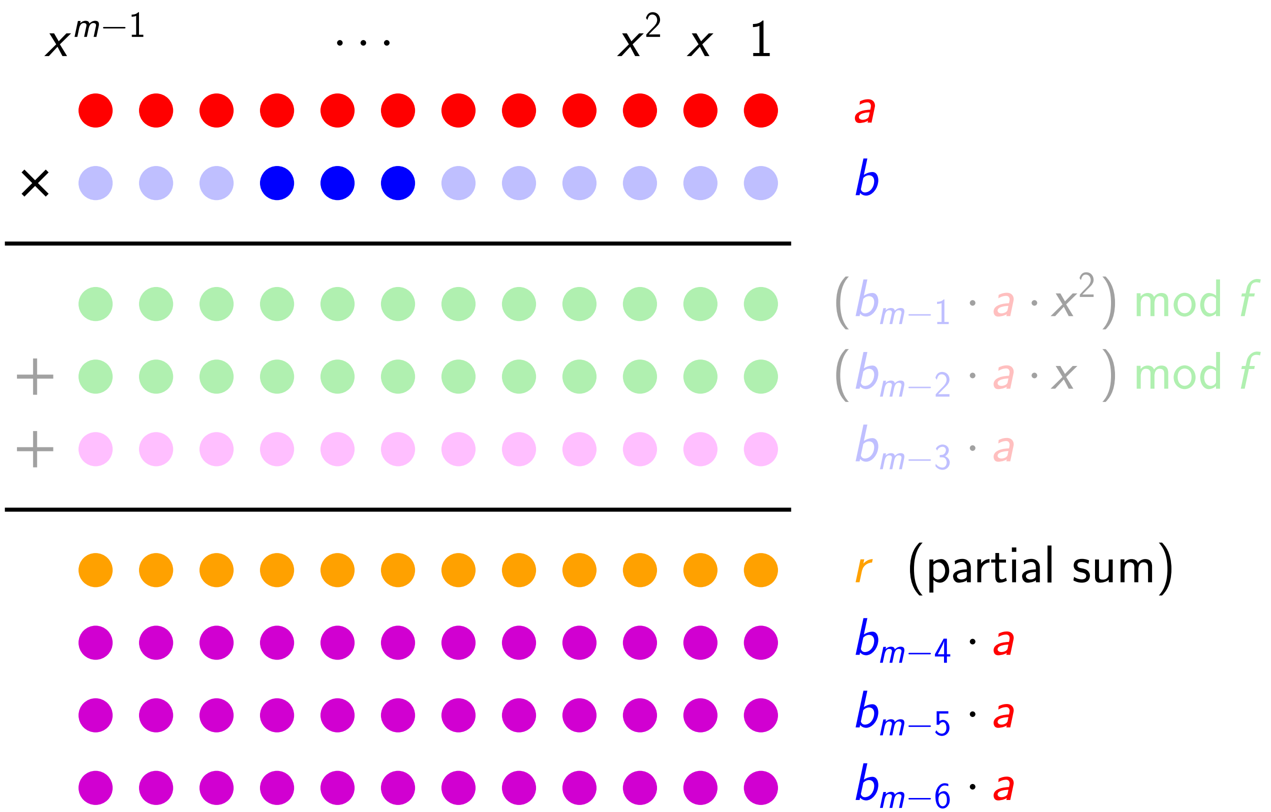
Multiplication over $\mathbb{F}_{3^m}$

- Example for $D = 3$ (3 coefficients per iteration):



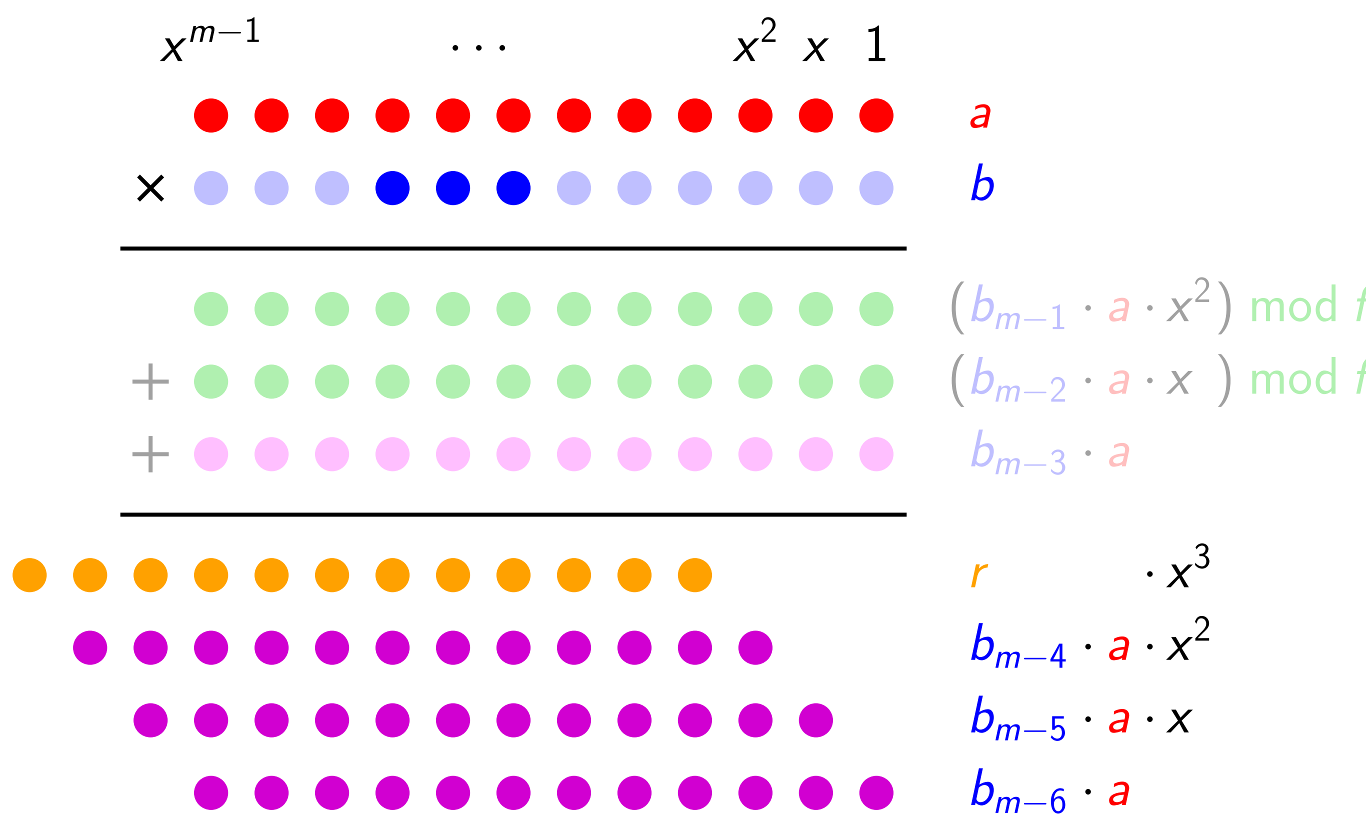
Multiplication over $\mathbb{F}_{3^m}$

- Example for $D = 3$ (3 coefficients per iteration):



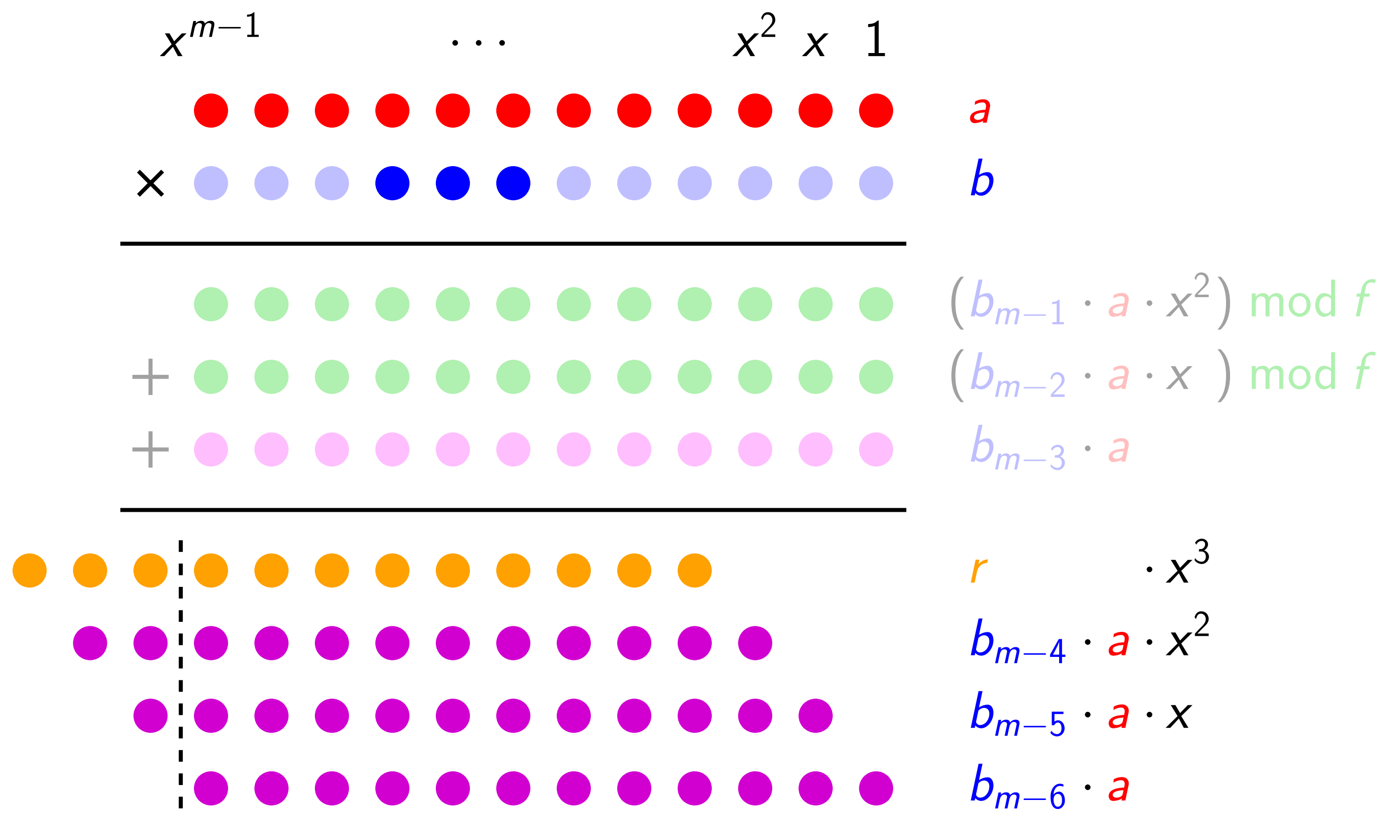
Multiplication over $\mathbb{F}_{3^m}$

- Example for $D = 3$ (3 coefficients per iteration):



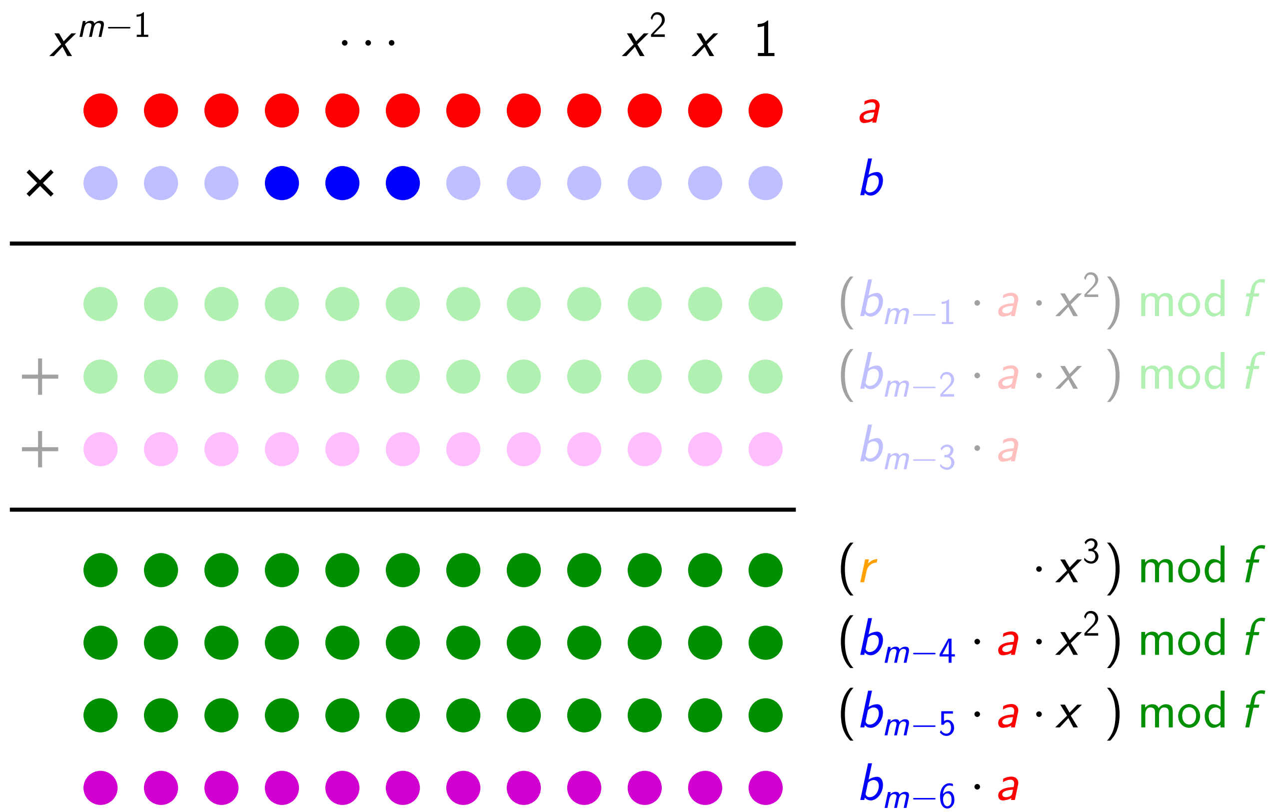
Multiplication over $\mathbb{F}_{3^m}$

- Example for $D = 3$ (3 coefficients per iteration):



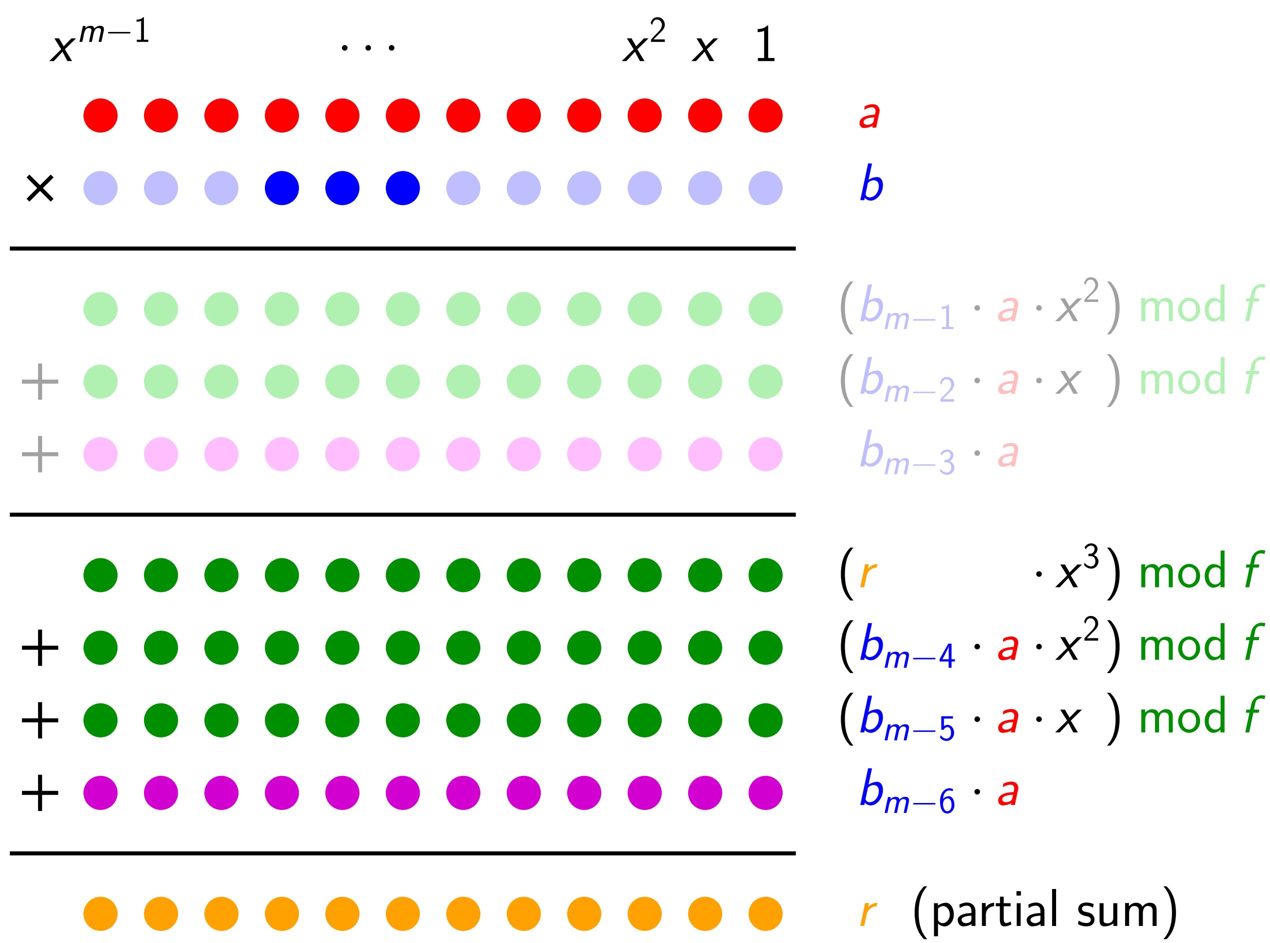
Multiplication over $\mathbb{F}_{3^m}$

- Example for $D = 3$ (3 coefficients per iteration):



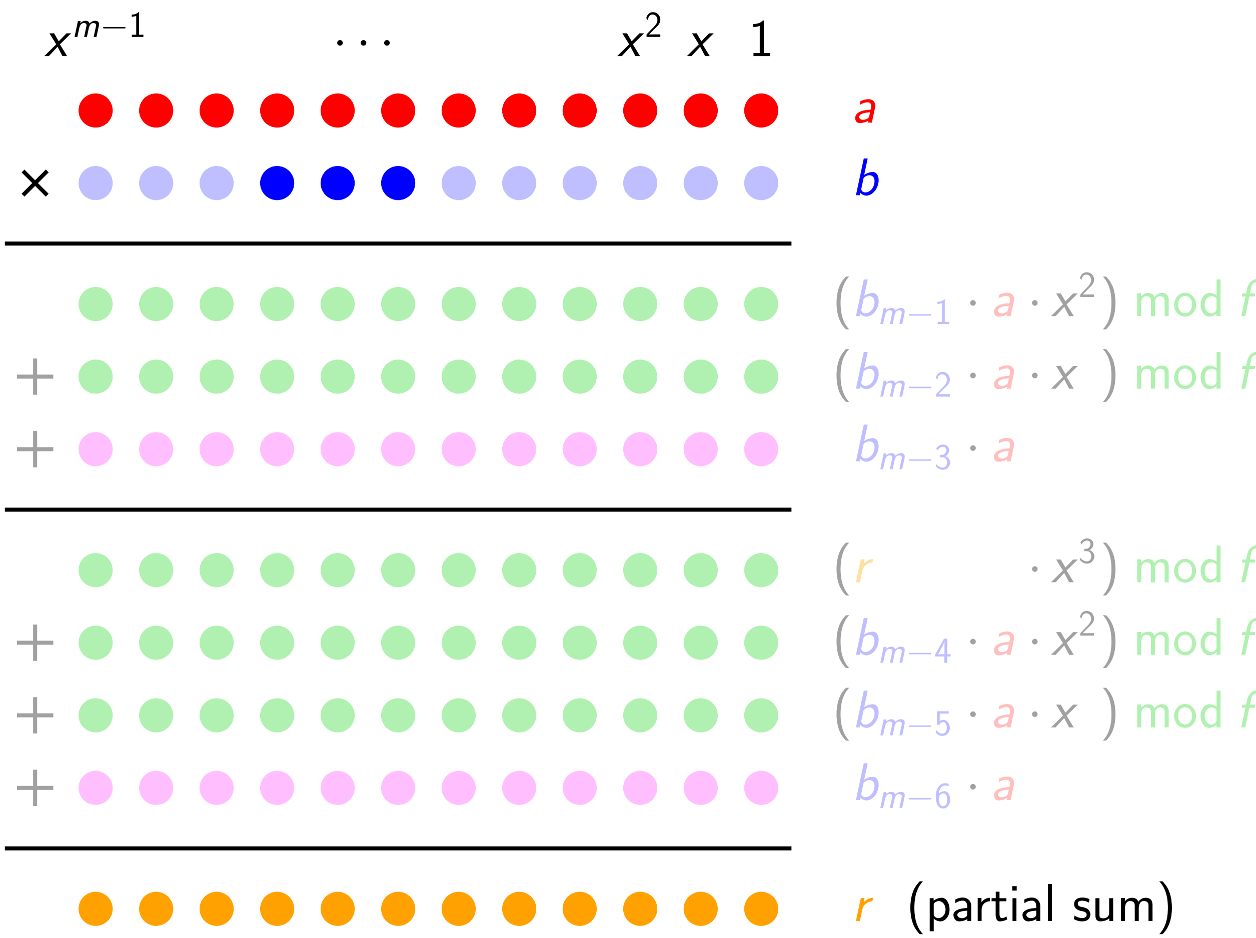
Multiplication over $\mathbb{F}_{3^m}$

- Example for $D = 3$ (3 coefficients per iteration):



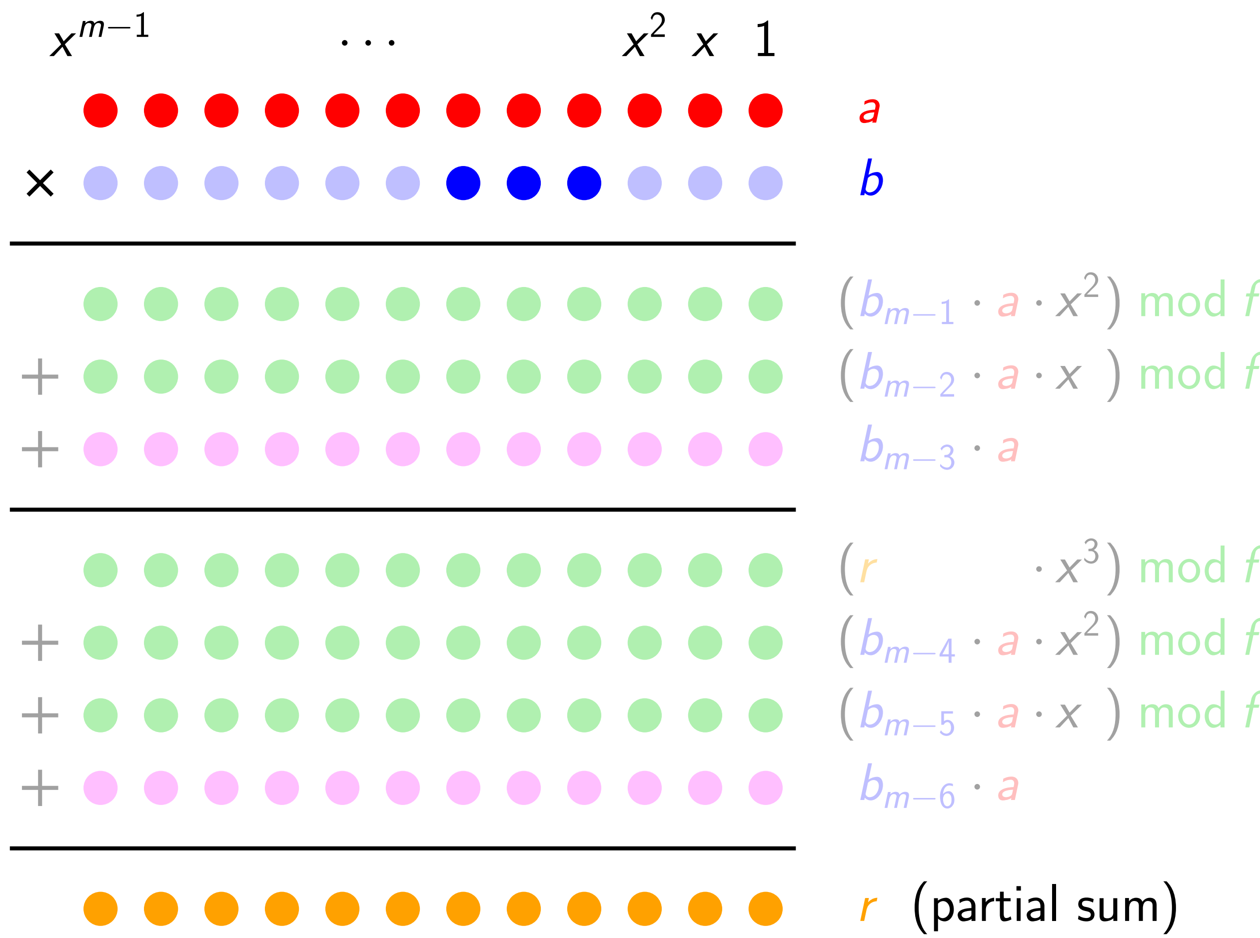
Multiplication over $\mathbb{F}_{3^m}$

- Example for $D = 3$ (3 coefficients per iteration):



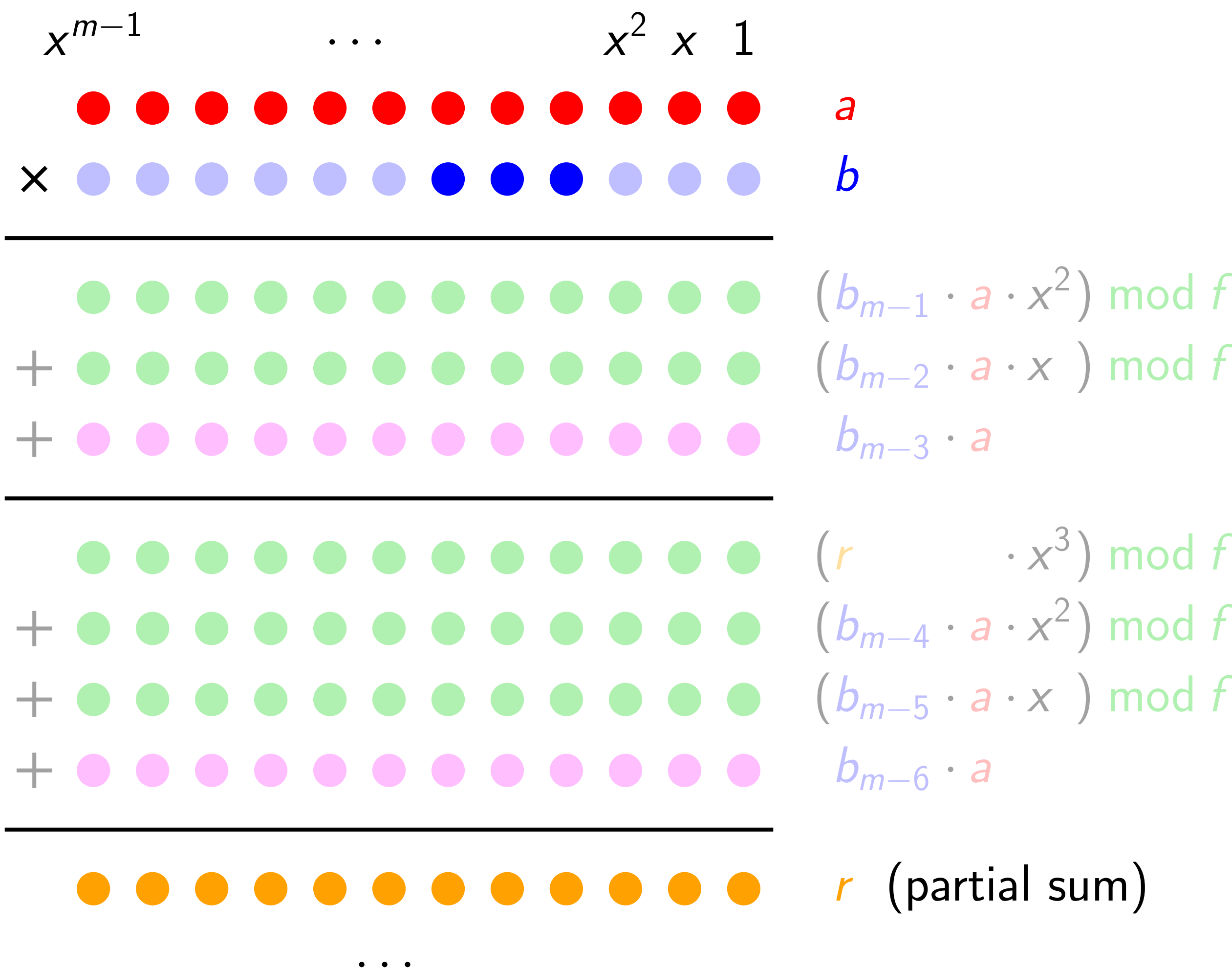
Multiplication over $\mathbb{F}_{3^m}$

- Example for $D = 3$ (3 coefficients per iteration):



Multiplication over $\mathbb{F}_{3^m}$

- Example for $D = 3$ (3 coefficients per iteration):



Multiplication over $\mathbb{F}_{3^m}$

- Computing the partial products $b_j \cdot a$:
  - coefficient-wise multiplication over $\mathbb{F}_3$: $(b_j \cdot a_i) \bmod 3$
  - multiplications over $\mathbb{F}_3$: small look-up tables

Multiplication over $\mathbb{F}_{3^m}$

- Computing the partial products $b_j \cdot a$:
  - coefficient-wise multiplication over $\mathbb{F}_3$: $(b_j \cdot a_i) \bmod 3$
  - multiplications over $\mathbb{F}_3$: small look-up tables
- Multiplication by x^j: simple shift (only wires)

Multiplication over $\mathbb{F}_{3^m}$

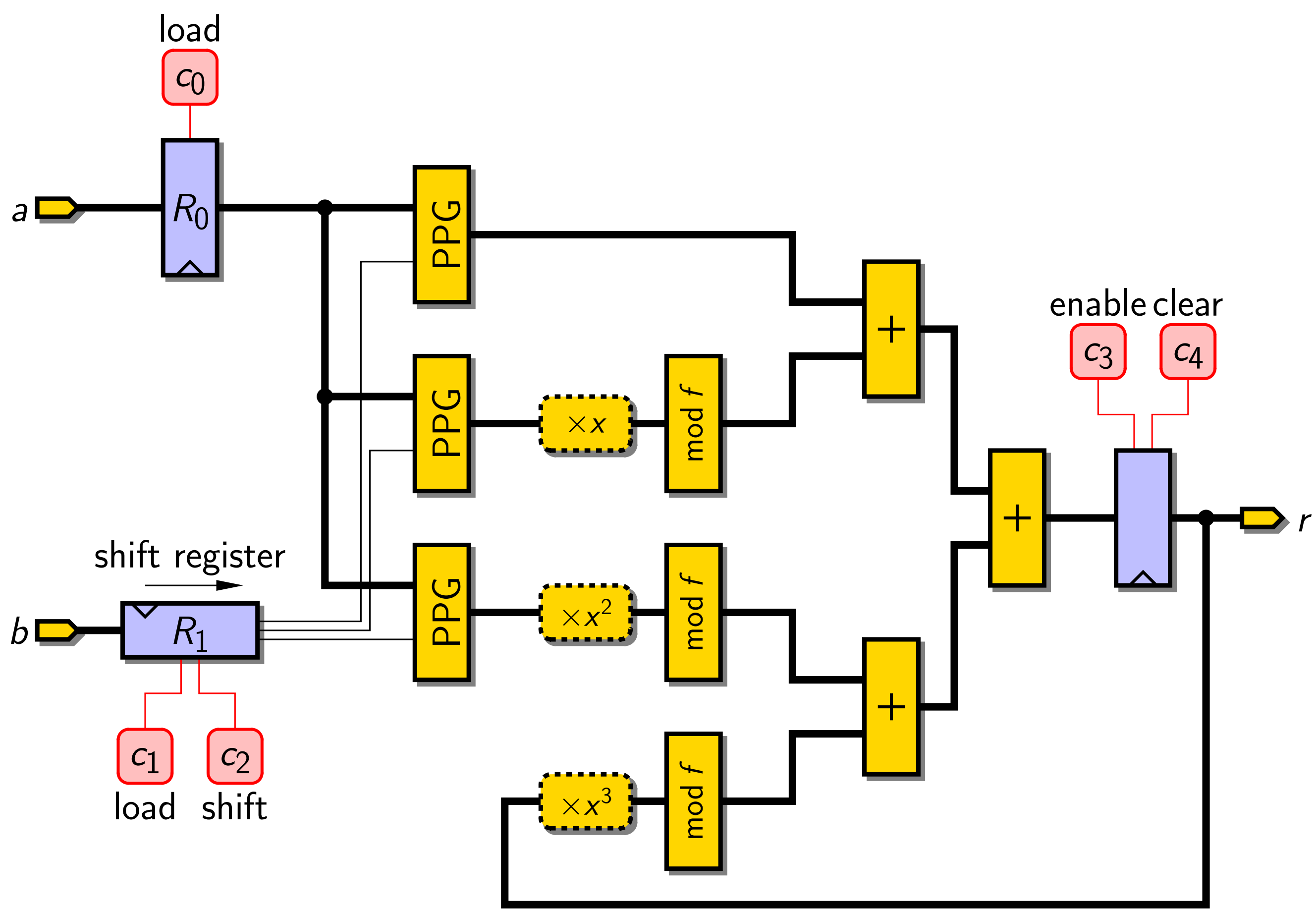
- Computing the partial products $b_j \cdot a$:
  - coefficient-wise multiplication over $\mathbb{F}_3$: $(b_j \cdot a_i) \bmod 3$
  - multiplications over $\mathbb{F}_3$: small look-up tables
- Multiplication by x^j: simple shift (only wires)
- Modulo f reduction:
  - $f = x^m + f_{m-1}x^{m-1} + \cdots + f_1 x + f_0$ gives
  $$x^m \equiv (-f_{m-1})x^{m-1} + \cdots + (-f_1)x + (-f_0) \pmod{f}$$
  - highest degree of polynomial to reduce: $m + D - 1$
  - if f is carefully selected (*e.g.* a trinomial or pentanomial), only a few multiplications and additions over $\mathbb{F}_3$

Multiplication over $\mathbb{F}_{3^m}$

- Computing the partial products $b_j \cdot a$:
  - coefficient-wise multiplication over $\mathbb{F}_3$: $(b_j \cdot a_i) \bmod 3$
  - multiplications over $\mathbb{F}_3$: small look-up tables
- Multiplication by x^j: simple shift (only wires)
- Modulo f reduction:
  - $f = x^m + f_{m-1}x^{m-1} + \cdots + f_1 x + f_0$ gives
$$x^m \equiv (-f_{m-1})x^{m-1} + \cdots + (-f_1)x + (-f_0) \pmod{f}$$
  - highest degree of polynomial to reduce: $m + D - 1$
  - if f is carefully selected (*e.g.* a trinomial or pentanomial), only a few multiplications and additions over $\mathbb{F}_3$
  - example for $m = 97$: $f = x^{97} + x^{12} + 2$

Multiplication over $\mathbb{F}_{3^m}$

- Example for $D = 3$ (3 coefficients per iteration):



Frobenius map over $\mathbb{F}_{3^m}$: cubing

- Since $\binom{3}{1} = \binom{3}{2} = 3$:

$$a^3 \equiv a_{m-1}x^{3(m-1)} + \cdots + a_1x^3 + a_0 \pmod 3$$

- Degree-$(3m-3)$ polynomial: requires a modulo f reduction

Frobenius map over $\mathbb{F}_{3^m}$: cubing

- Since $\binom{3}{1} = \binom{3}{2} = 3$:

$$a^3 \equiv a_{m-1}x^{3(m-1)} + \cdots + a_1x^3 + a_0 \pmod 3$$

- Degree-$(3m-3)$ polynomial: requires a modulo f reduction
- Symbolic computation of the reduction:
  each coefficient of the result is a linear combination of the a_i's

$$a^3 \bmod f = \sum_{j=0}^{n-1} w_j \cdot \mu_j$$

with $w_j \in \mathbb{F}_3$, $\mu_j \in \mathbb{F}_{3^m}$, and $\mu_{j,i} \in \{0\} \cup \{a_{m-1}, \ldots, a_1, a_0\}$

Frobenius map over $\mathbb{F}_{3^m}$

- Example for $m = 97$ and $f = x^{97} + x^{12} + 2$:

$$
\begin{array}{rcl}
a^3 \bmod f & = & (a_{32}x^{96} + a_{64}x^{95} + a_{96}x^{94} + \cdots + a_{33}x^2 + a_{65}x + a_0) \times 1 \\
& + & (\ 0 + 0 + a_{88}x^{94} + \cdots + 0 + 0 + a_{89}) \times 1 \\
& + & (\ 0 + 0 + a_{92}x^{94} + \cdots + 0 + 0 + a_{93}) \times 1 \\
& + & (\ 0 + a_{60}x^{95} + 0 + \cdots + 0 + a_{61}x + 0\ ) \times 2
\end{array}
$$

Frobenius map over $\mathbb{F}_{3^m}$

- Example for $m = 97$ and $f = x^{97} + x^{12} + 2$:

$$
\begin{array}{rcl}
a^3 \bmod f & = & (a_{32}x^{96} + a_{64}x^{95} + a_{96}x^{94} + \cdots + a_{33}x^2 + a_{65}x + a_0) \times 1 \\
& + & (0 + 0 + a_{88}x^{94} + \cdots + 0 + 0 + a_{89}) \times 1 \\
& + & (0 + 0 + a_{92}x^{94} + \cdots + 0 + 0 + a_{93}) \times 1 \\
& + & (0 + a_{60}x^{95} + 0 + \cdots + 0 + a_{61}x + 0) \times 2 \\
& = & (a_{32}x^{96} + a_{64}x^{95} + a_{96}x^{94} + \cdots + a_{33}x^2 + a_{65}x + a_0) \times 1 \\
& + & (0 + a_{60}x^{95} + a_{88}x^{94} + \cdots + 0 + a_{61}x + a_{89}) \times 1 \\
& + & (0 + a_{60}x^{95} + a_{92}x^{94} + \cdots + 0 + a_{61}x + a_{93}) \times 1
\end{array}
$$

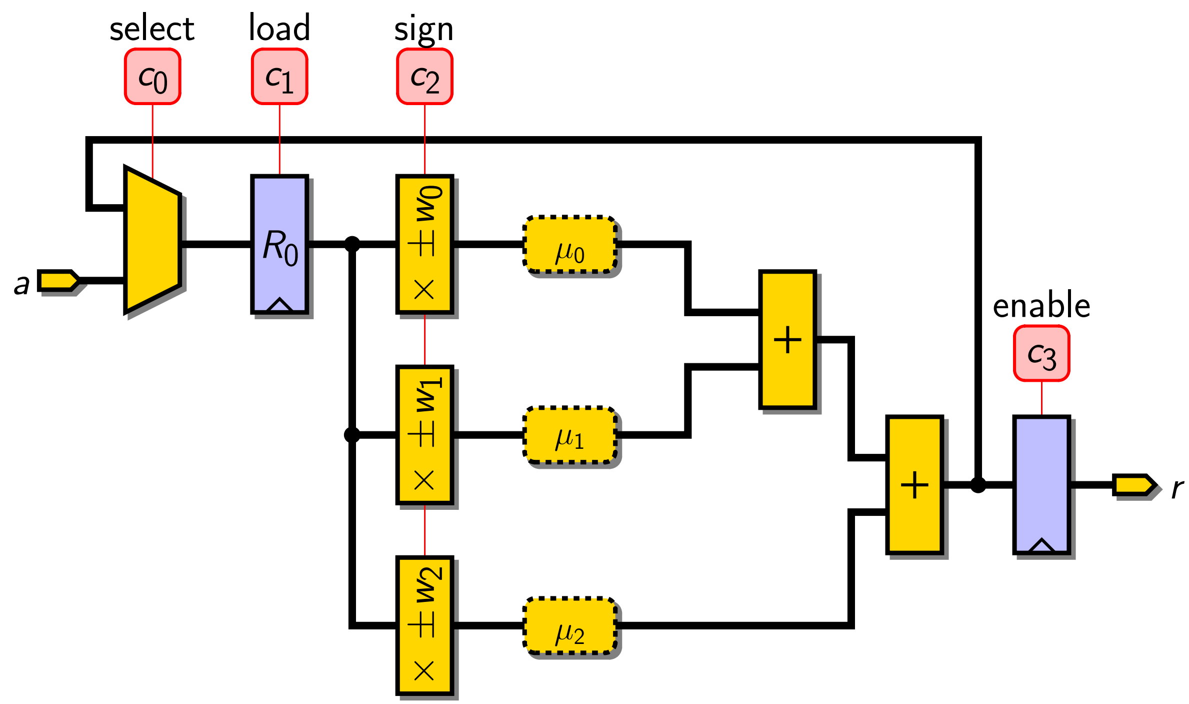
Frobenius map over $\mathbb{F}_{3^m}$

- Example for $m = 97$ and $f = x^{97} + x^{12} + 2$:

$$
\begin{array}{rcl}
a^3 \bmod f & = & (a_{32}x^{96} + a_{64}x^{95} + a_{96}x^{94} + \cdots + a_{33}x^2 + a_{65}x + a_0) \times 1 \\
& + & (\; 0 + 0 + a_{88}x^{94} + \cdots + 0 + 0 + a_{89}) \times 1 \\
& + & (\; 0 + 0 + a_{92}x^{94} + \cdots + 0 + 0 + a_{93}) \times 1 \\
& + & (\; 0 + a_{60}x^{95} + 0 + \cdots + 0 + a_{61}x + 0) \times 2 \\
& = & (a_{32}x^{96} + a_{64}x^{95} + a_{96}x^{94} + \cdots + a_{33}x^2 + a_{65}x + a_0) \times 1 \\
& + & (\; 0 + a_{60}x^{95} + a_{88}x^{94} + \cdots + 0 + a_{61}x + a_{89}) \times 1 \\
& + & (\; 0 + a_{60}x^{95} + a_{92}x^{94} + \cdots + 0 + a_{61}x + a_{93}) \times 1
\end{array}
$$

- Required hardware:
  - only wires to compute the μ_j's
  - multiplications over $\mathbb{F}_3$ for the weights w_j
  - multi-operand addition over $\mathbb{F}_{3^m}$

Frobenius map over $\mathbb{F}_{3^m}$



- feedback loop for successive cubings
- sign selection for computing either a^3 or $-a^3$

Inversion over $\mathbb{F}_{3^m}$

- Extended Euclidean Algorithm?

Inversion over $\mathbb{F}_{3^m}$

- Extended Euclidean Algorithm?
  - fast computation
  - ... but need for additional hardware

Inversion over $\mathbb{F}_{3^m}$

- Extended Euclidean Algorithm?
  - fast computation
  - ... but need for additional hardware
- Our solution: Fermat's little theorem

$$a^{-1} = a^{3^m-2} \qquad \text{on } \mathbb{F}_{3^m} \ (a \neq 0)$$

Inversion over $\mathbb{F}_{3^m}$

- Extended Euclidean Algorithm?
  - fast computation
  - ... but need for additional hardware
- Our solution: Fermat's little theorem

$$a^{-1} = a^{3^m - 2} \qquad \text{on } \mathbb{F}_{3^m} \ (a \neq 0)$$

  - algorithm by Itoh and Tsujii
  - requires only multiplications and cubings over $\mathbb{F}_{3^m}$

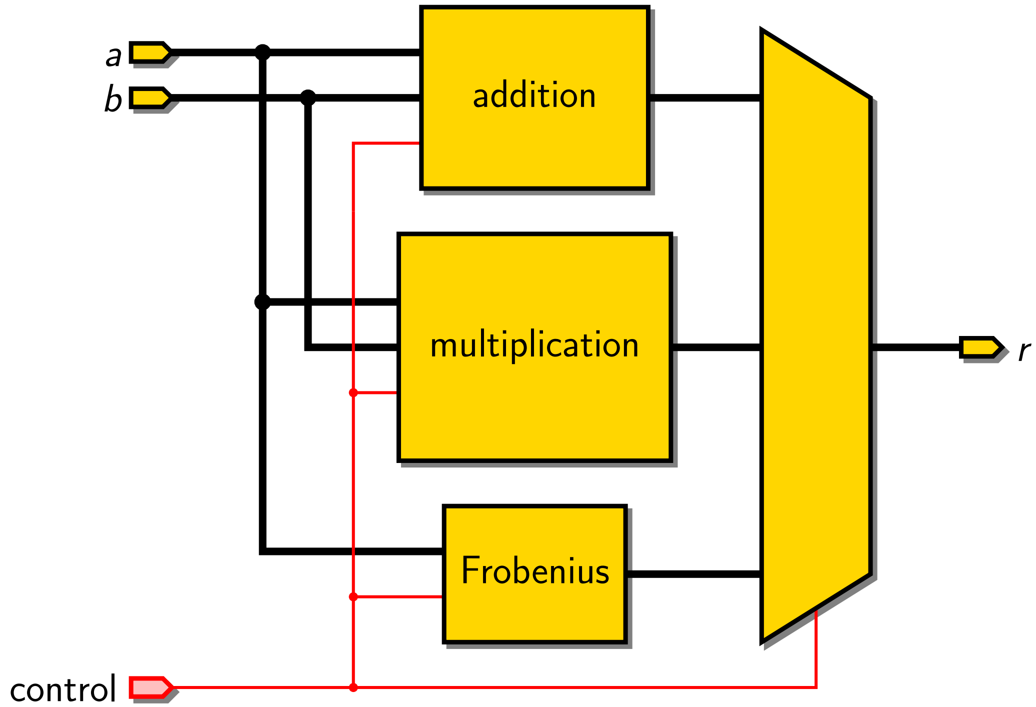
Inversion over $\mathbb{F}_{3^m}$

- Extended Euclidean Algorithm?
  - fast computation
  - ... but need for additional hardware
- Our solution: Fermat's little theorem

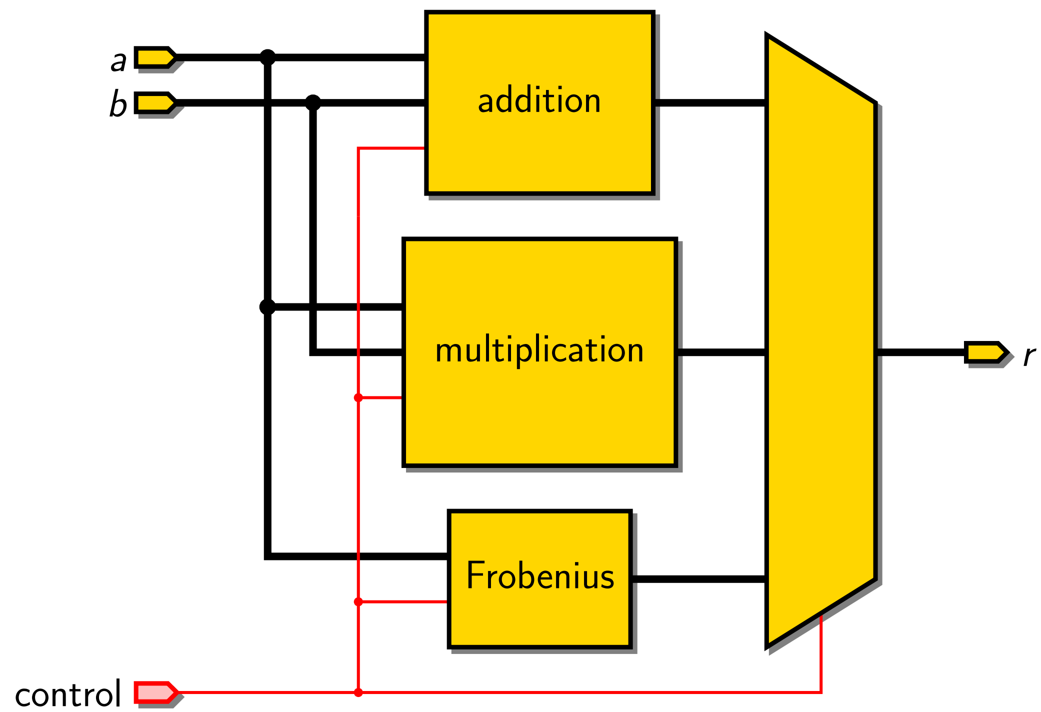
$$a^{-1} = a^{3^m-2} \qquad \text{on } \mathbb{F}_{3^m}\ (a \neq 0)$$

  - algorithm by Itoh and Tsujii
  - requires only multiplications and cubings over $\mathbb{F}_{3^m}$
  - only one inversion for the full pairing: delay overhead is negligible ($< 1\%$)

The full processing element



The full processing element



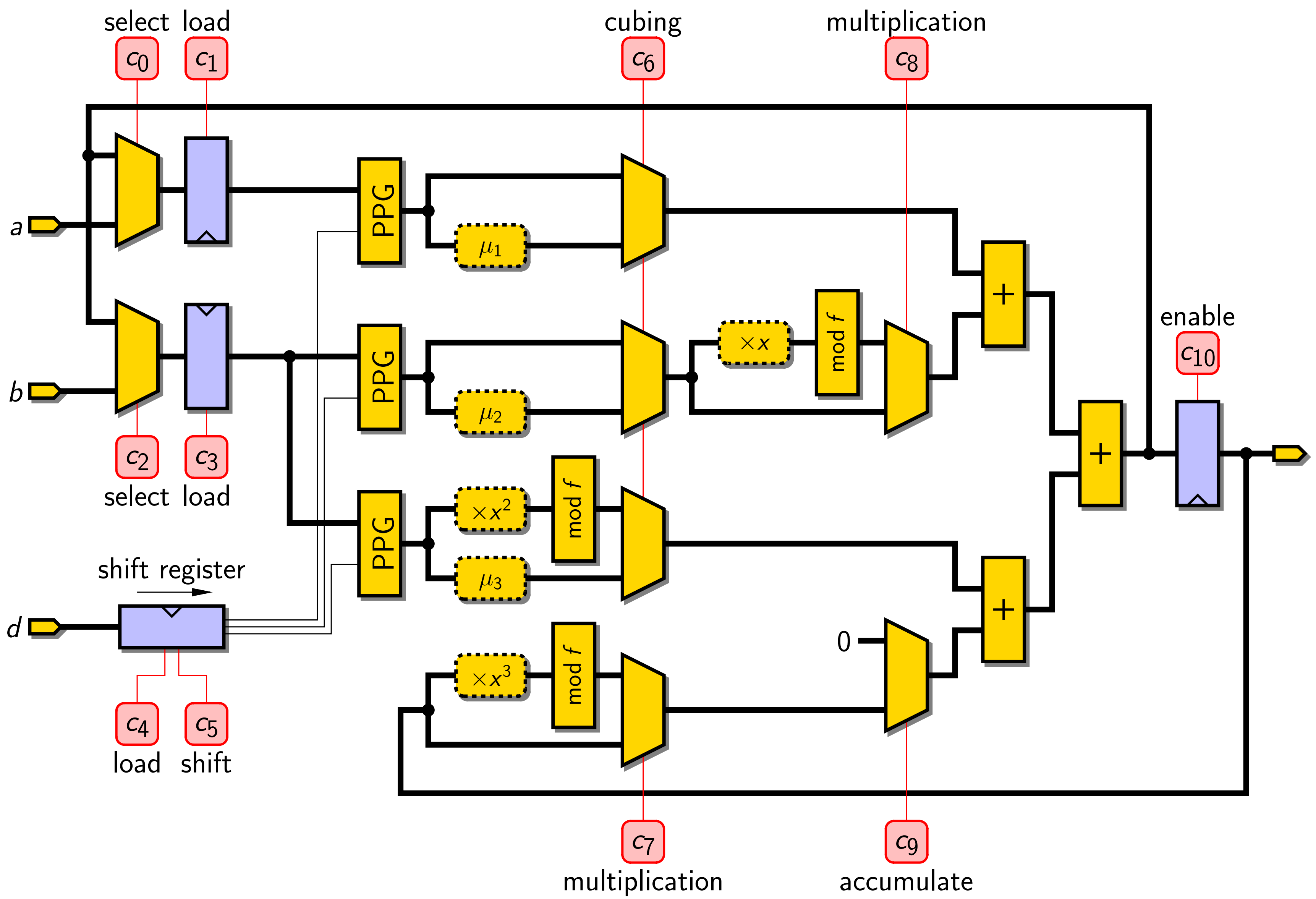
- For the Tate pairing:
  limited parallelism between additions, multiplications and Frobenius maps

- Can we share hardware resources between the three operators?

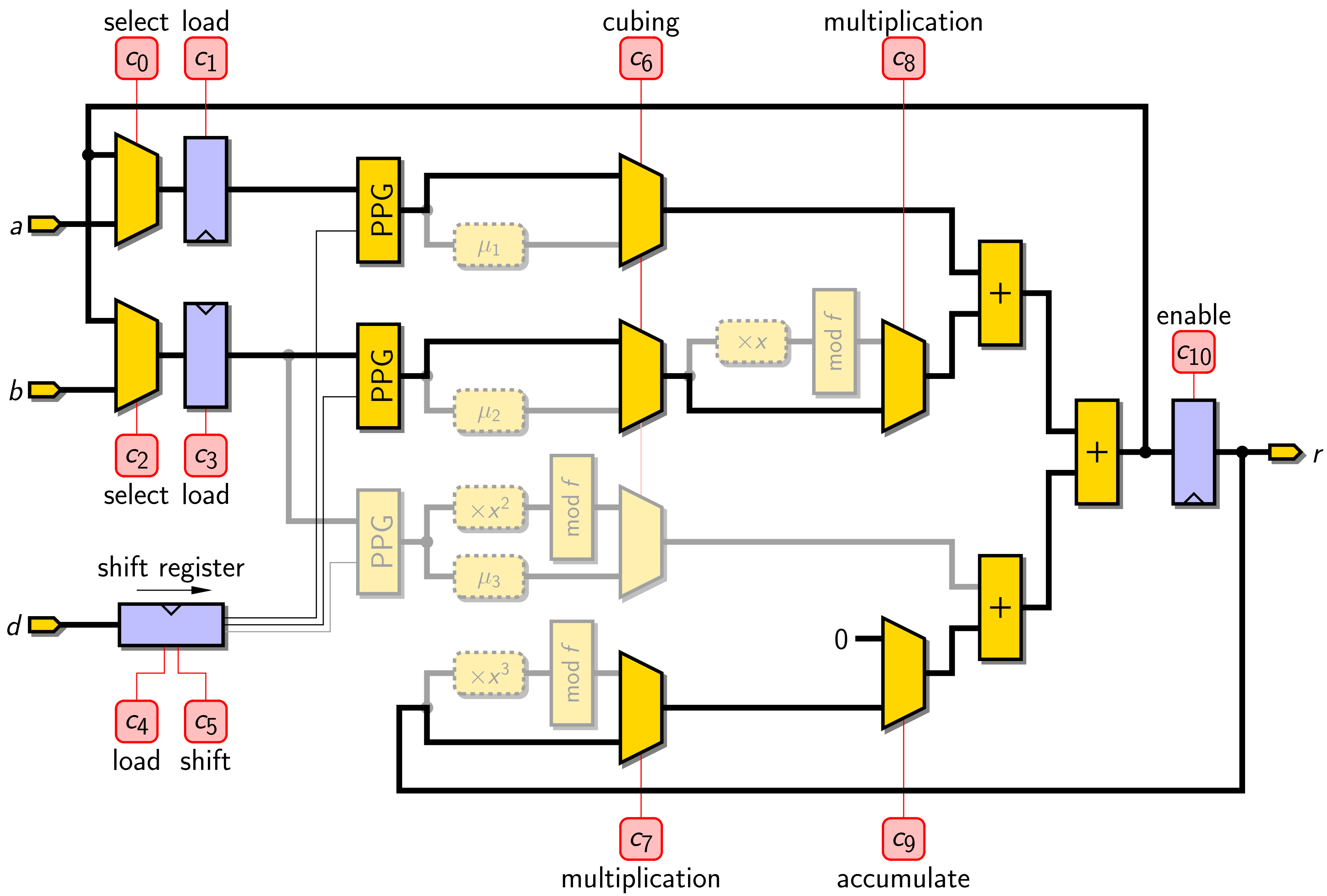
What can we share?

- Input and output registers
- Partial product generators:
  - sign selection for the addition / subtraction
  - partial products for the multiplication
  - multiplication by the w_j's for the Frobenius map
- Multi-operand addition tree
- Feedback loops for accumulation

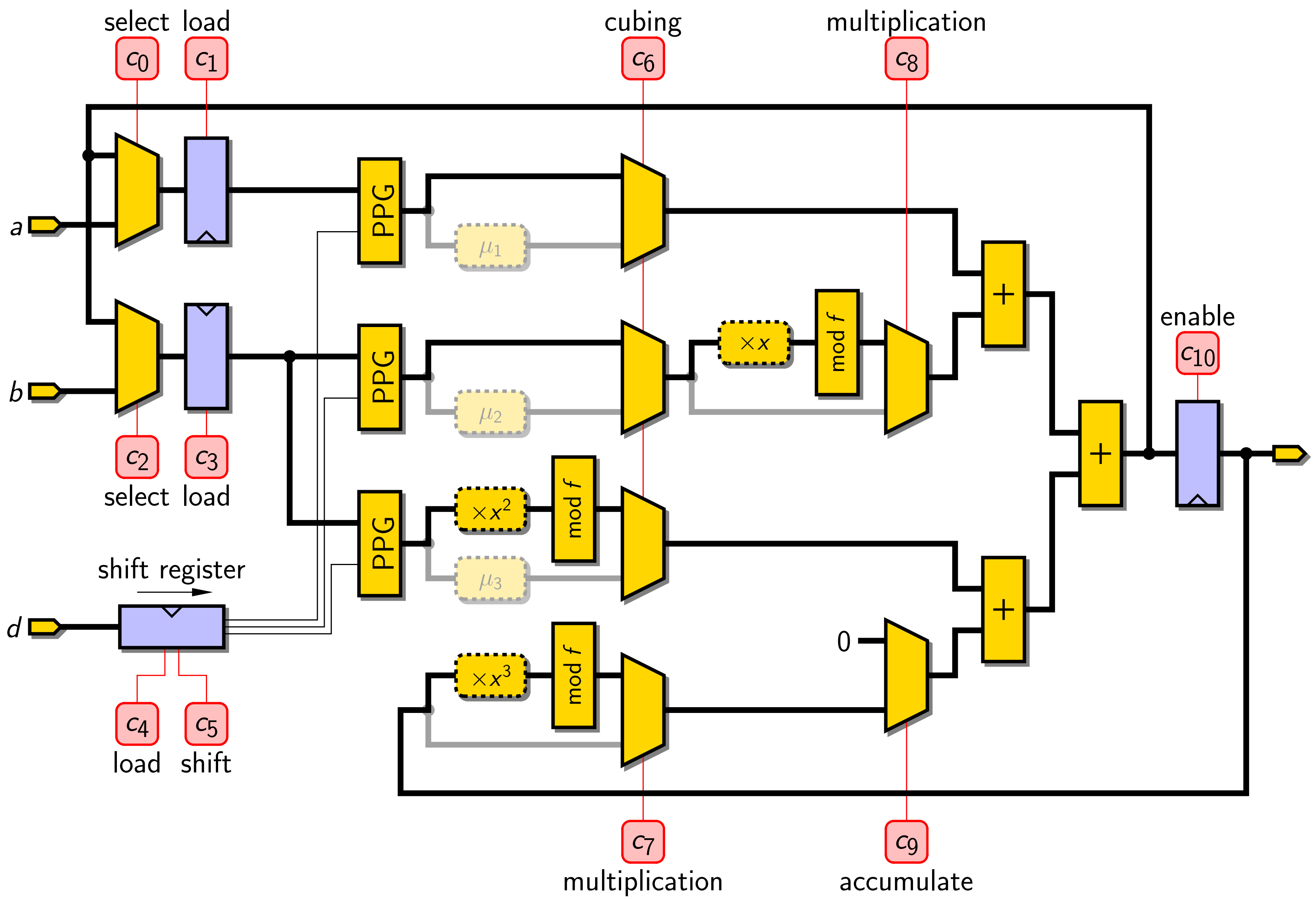
Our unified operator



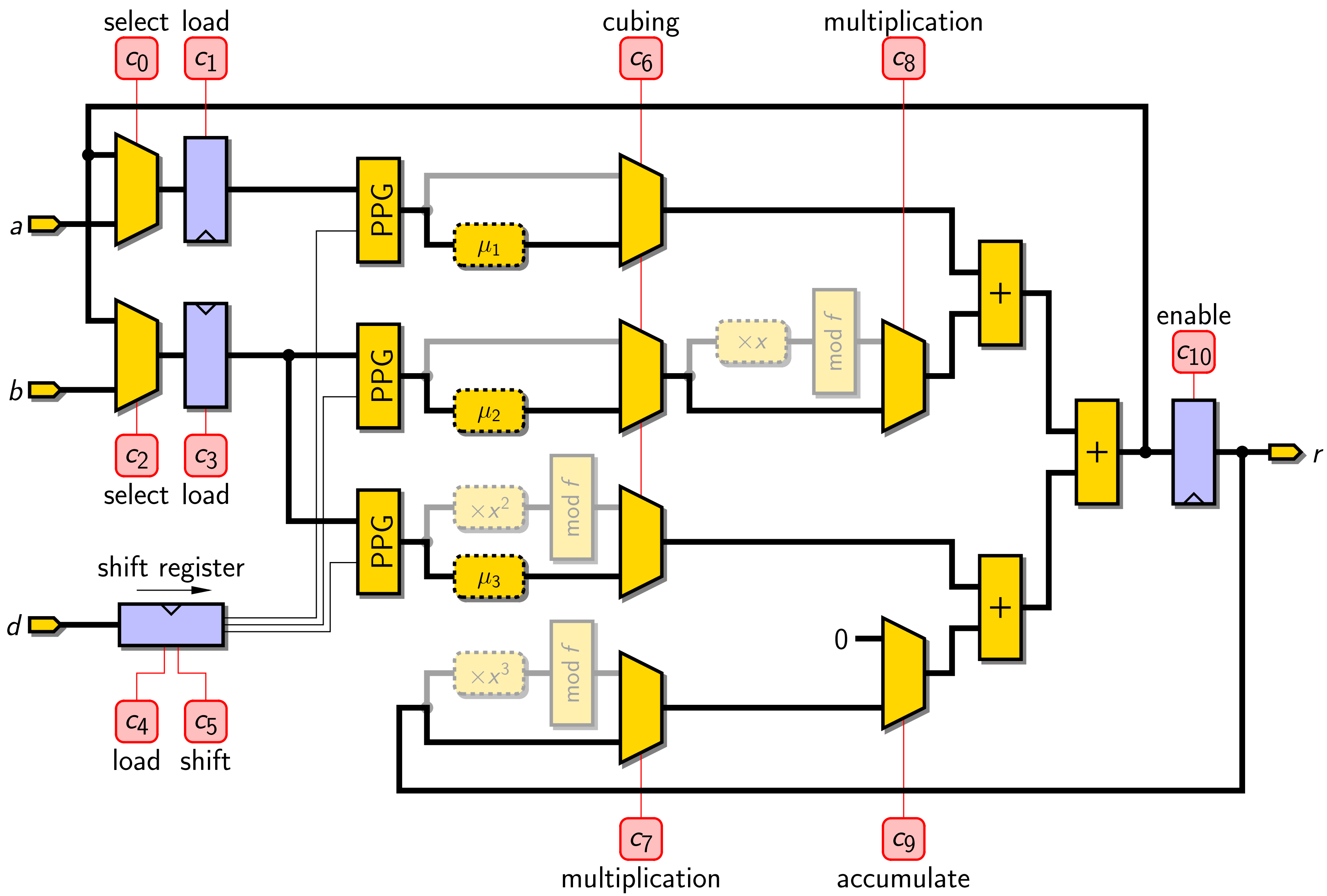
Our unified operator



Our unified operator



Our unified operator



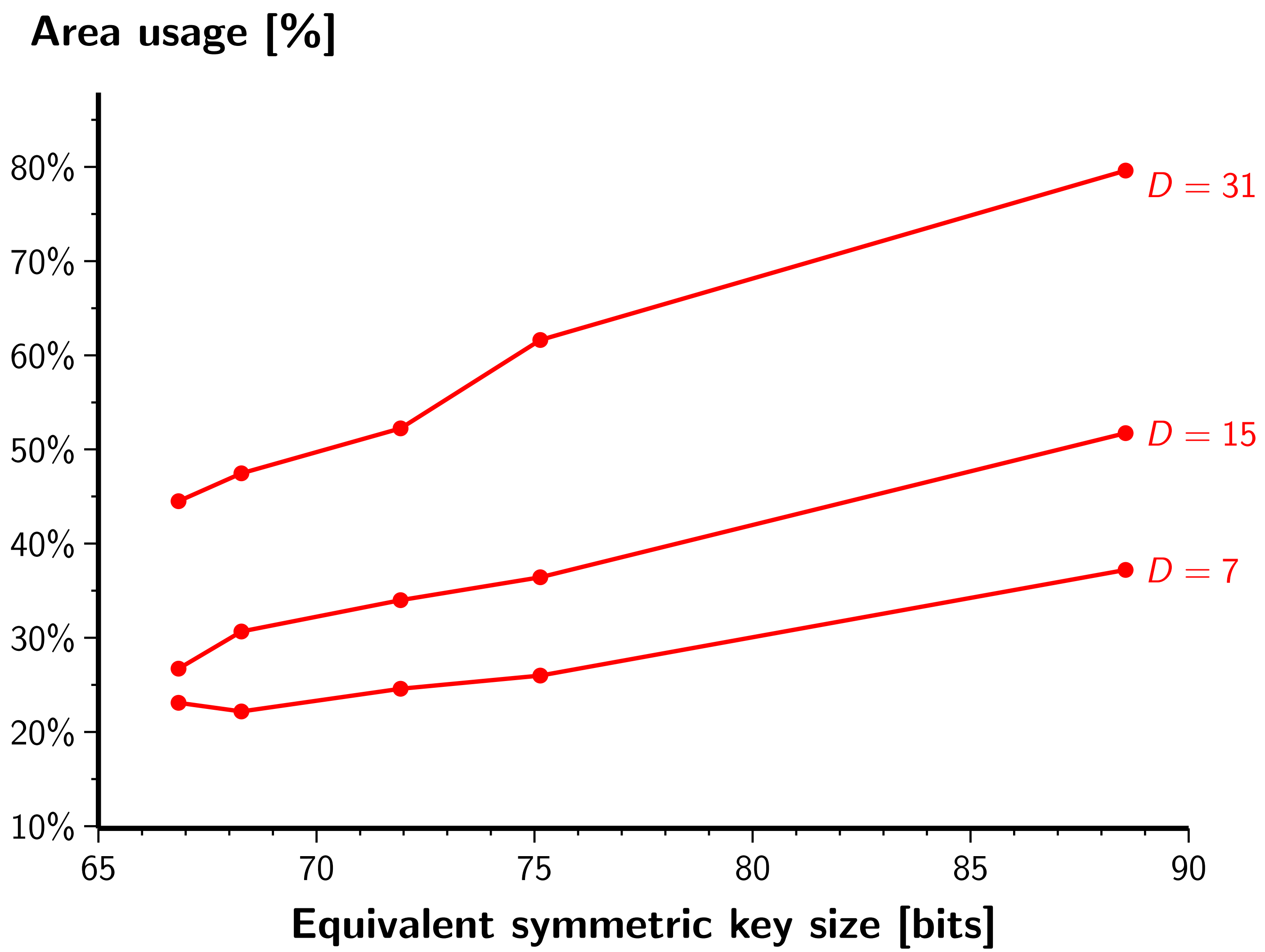
Outline of the talk

- Pairing-based cryptography
- Pairings over elliptic curves
- Finite-field arithmetic
- Implementation results
- Concluding thoughts

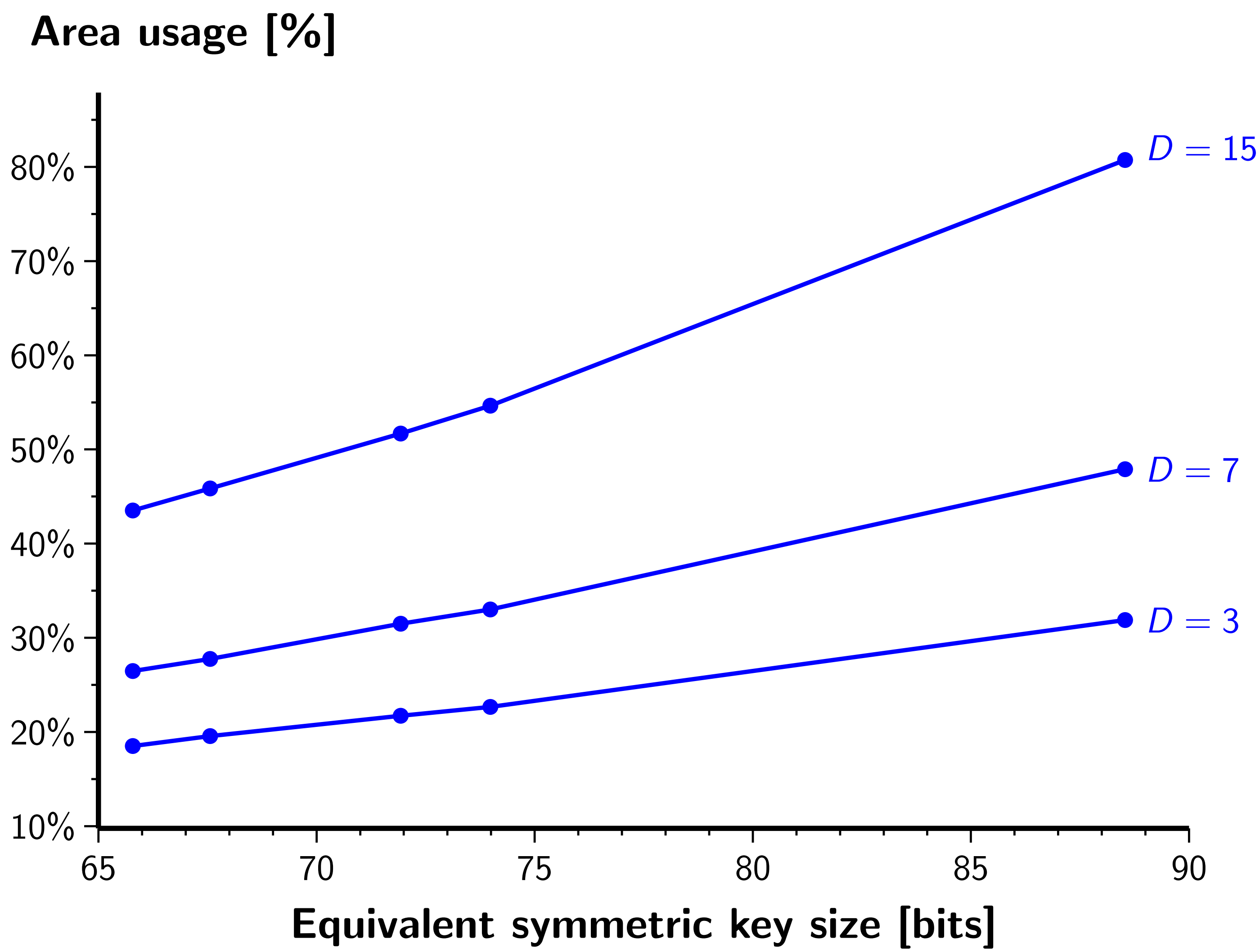
Experimental setup

- Full coprocessor for computation of the Tate pairing
- Architecture based on our unified operator
- Prototyped on a Xilinx Virtex-II Pro 20 FPGA (mid-range model)
- Post place-and-route results: area, computation time, AT product

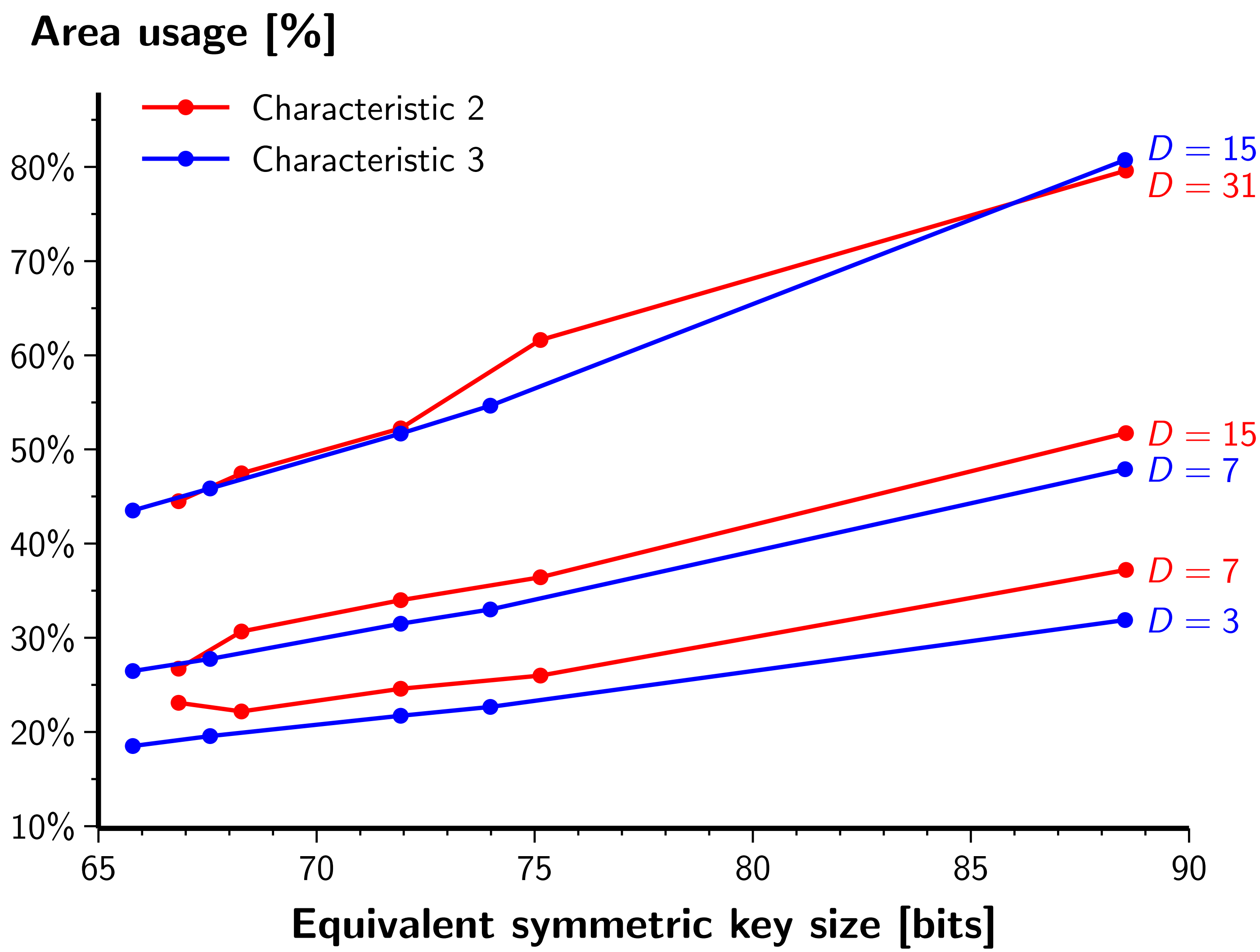
Coprocessor area (characteristic 2)



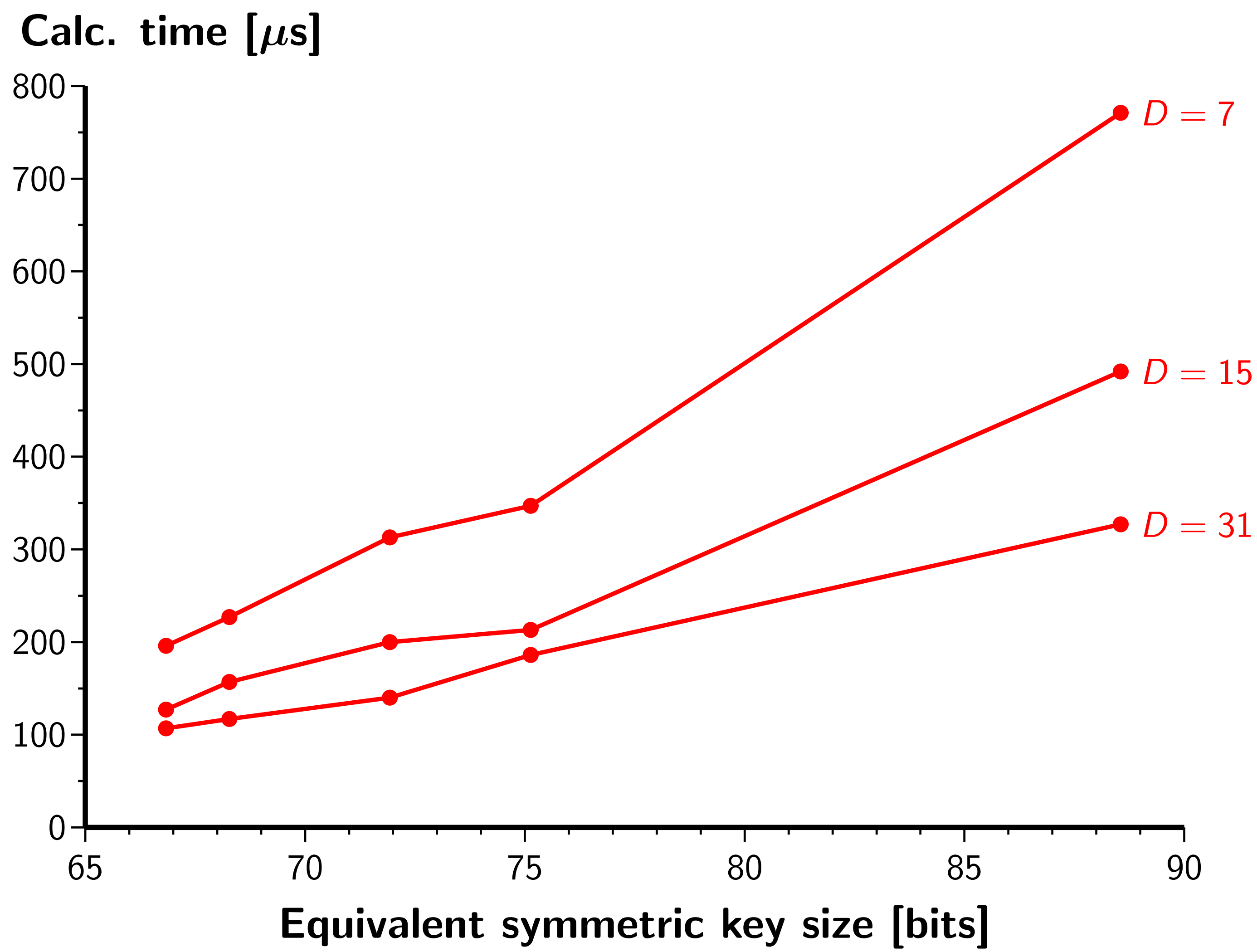
Coprocessor area (characteristic 3)



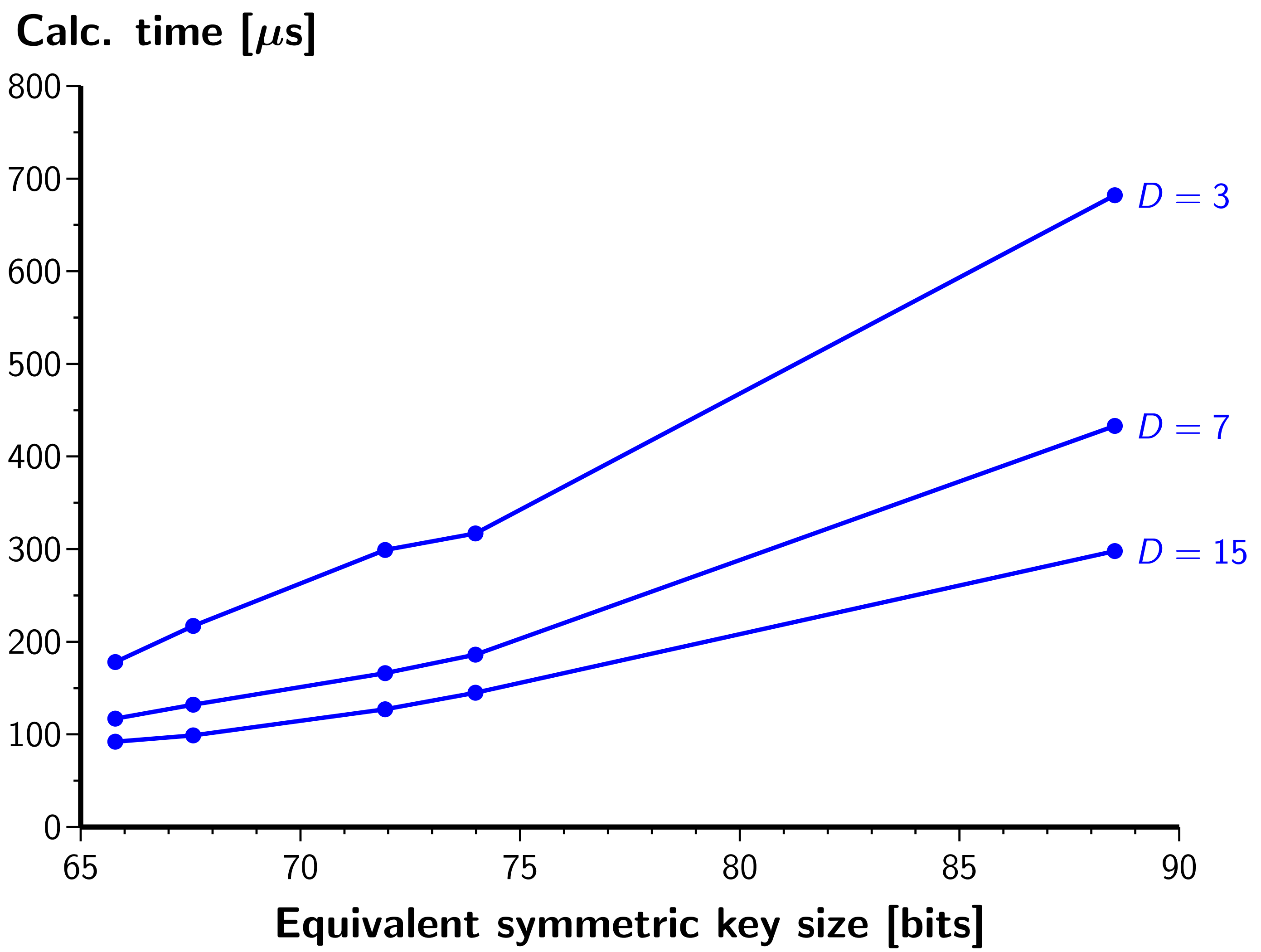
Coprocessor area



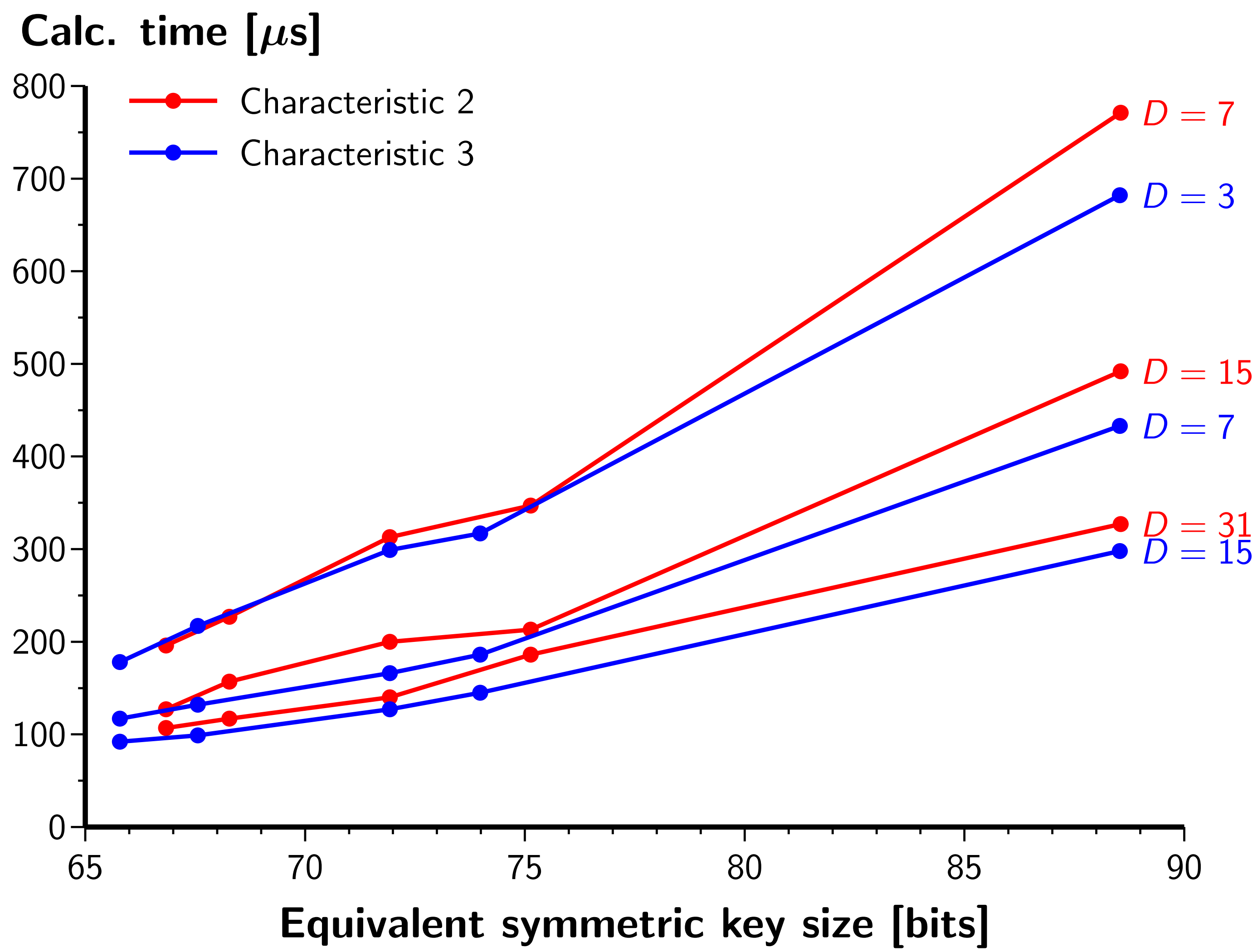
Calculation time (characteristic 2)



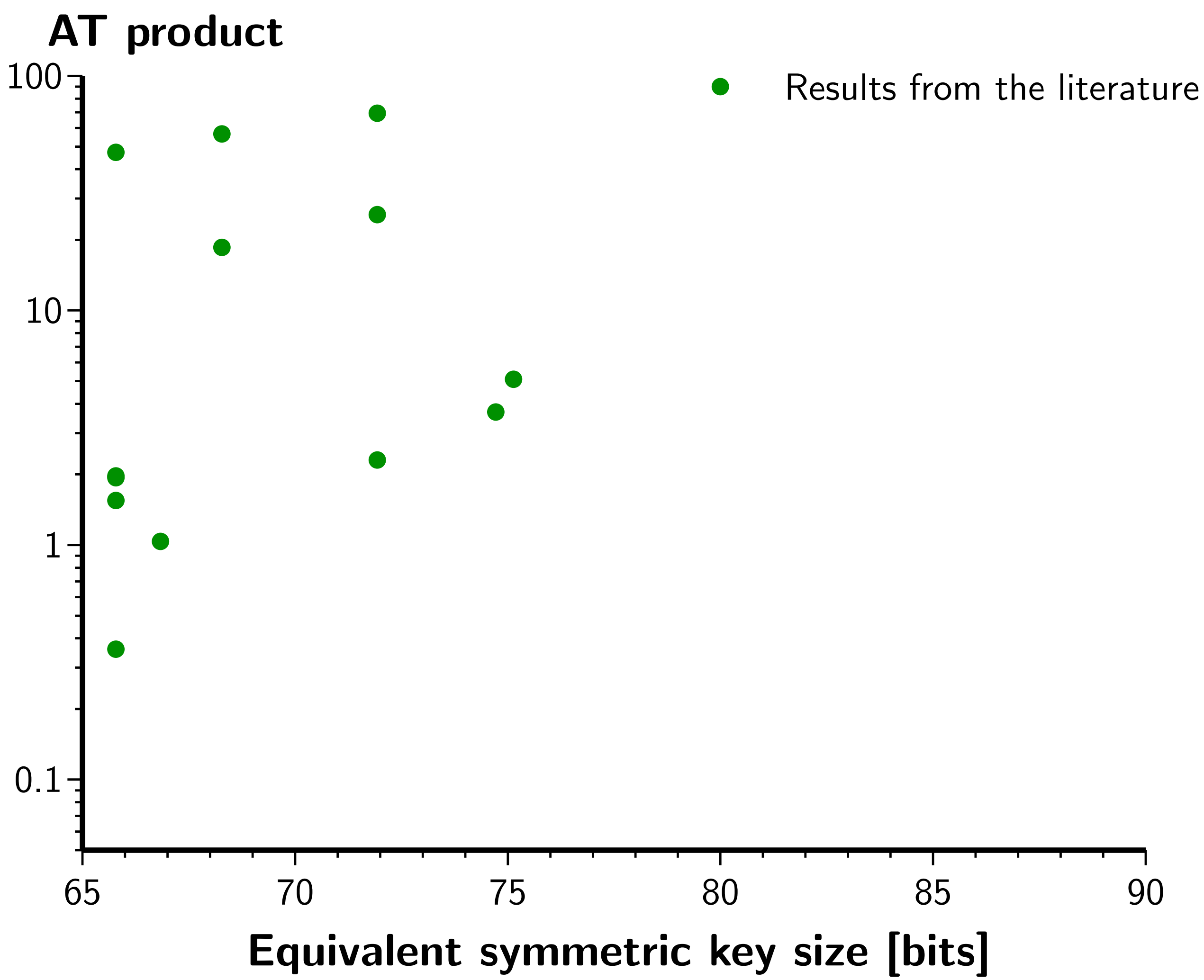
Calculation time (characteristic 3)



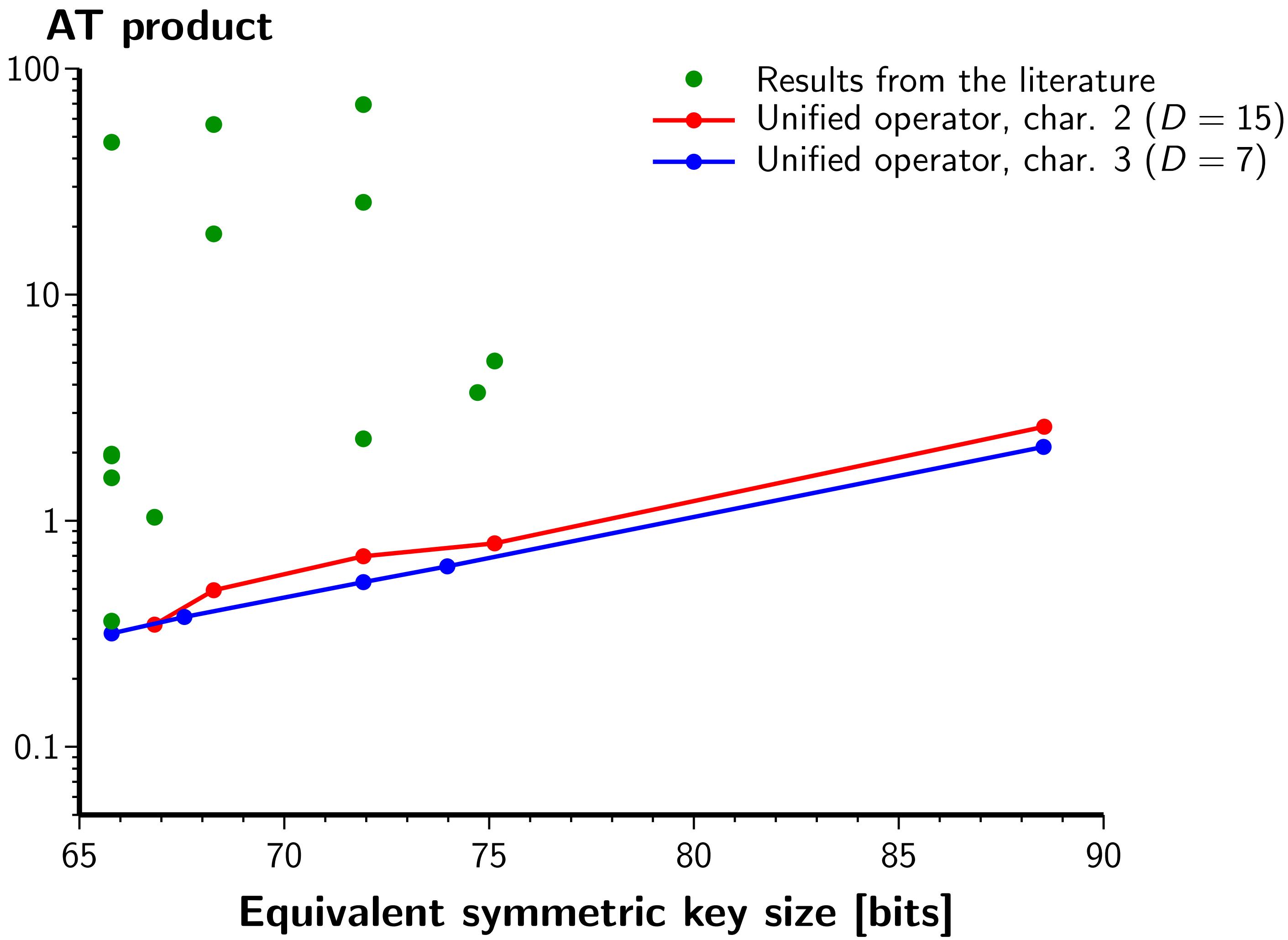
Calculation time



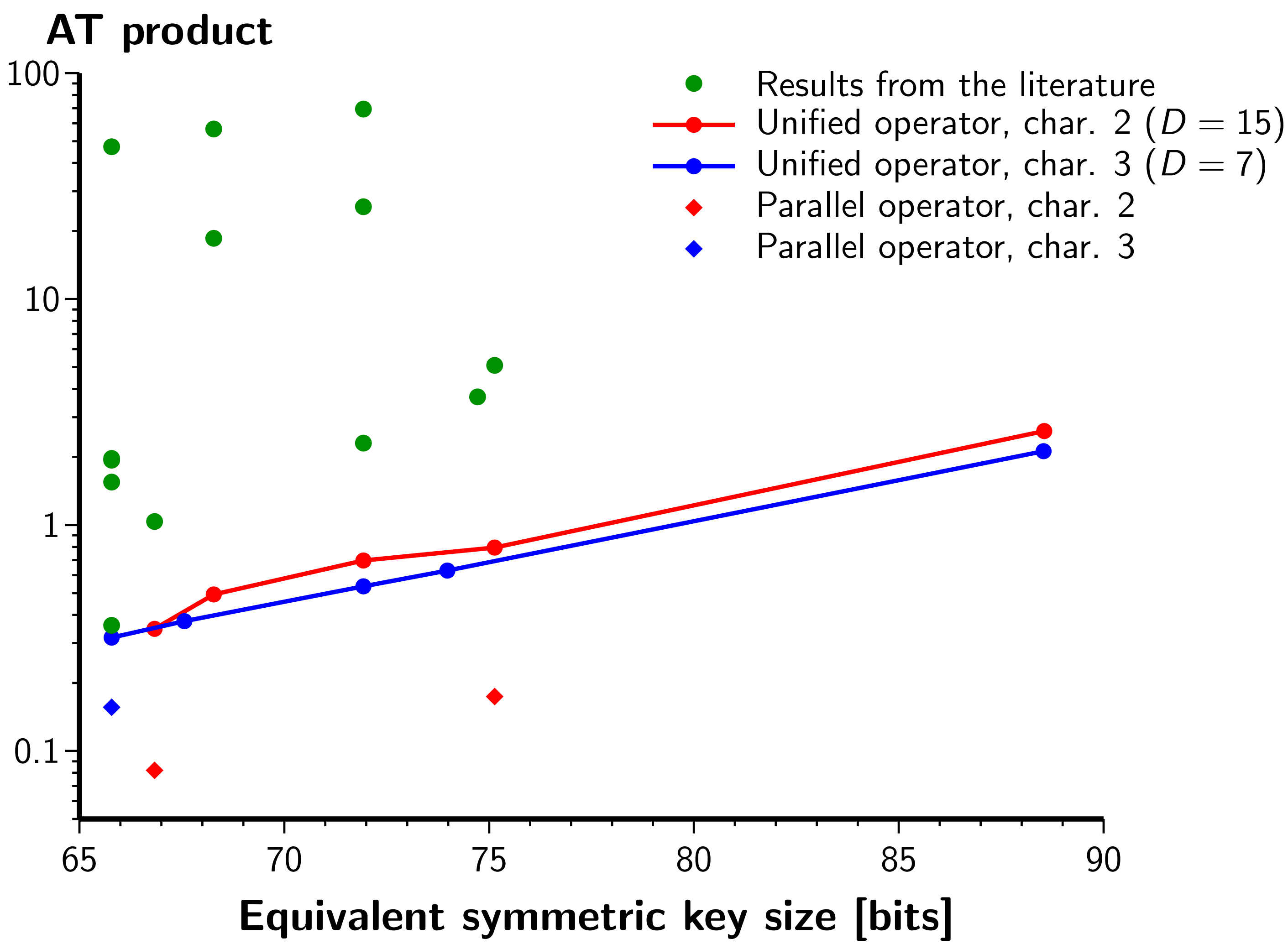
Comparison with published results



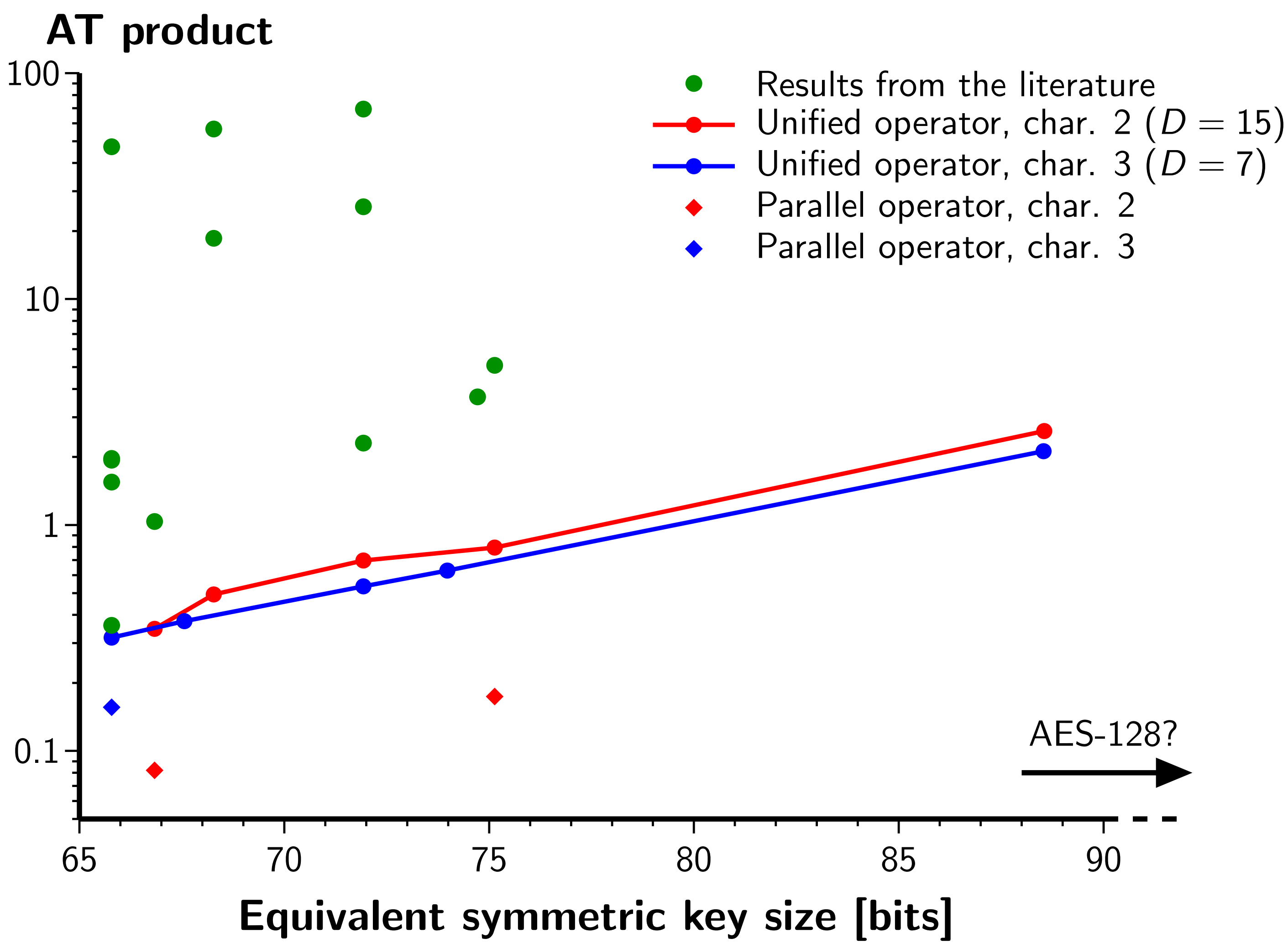
Comparison with published results



Comparison with published results



Comparison with published results



Outline of the talk

- Pairing-based cryptography
- Pairings over elliptic curves
- Finite-field arithmetic
- Implementation results
- Concluding thoughts

Concluding thoughts

- Characteristic 3 performs slightly better than characteristic 2
  - at least on our unified architecture
  - good overall performances vouch for stronger confidence in this observation

Concluding thoughts

- Characteristic 3 performs slightly better than characteristic 2
  - at least on our unified architecture
  - good overall performances vouch for stronger confidence in this observation
  - not true anymore on parallel architectures: the battle is not over!

Concluding thoughts

- Characteristic 3 performs slightly better than characteristic 2
  - at least on our unified architecture
  - good overall performances vouch for stronger confidence in this observation
  - not true anymore on parallel architectures: the battle is not over!
- Unified operator
  - small but also competitively fast
  - parameter D to explore the area-time tradeoff
  - high scalability: support for larger extension degrees and higher levels of security
  - automatic VHDL generation: ultra-fast development

Concluding thoughts

- Characteristic 3 performs slightly better than characteristic 2
  - at least on our unified architecture
  - good overall performances vouch for stronger confidence in this observation
  - not true anymore on parallel architectures: the battle is not over!
- Unified operator
  - small but also competitively fast
  - parameter D to explore the area-time tradeoff
  - high scalability: support for larger extension degrees and higher levels of security
  - automatic VHDL generation: ultra-fast development
- Perspectives
  - parallel architectures (work in progress with N. Cortez-Duarte and N. Estibals)
  - hyperelliptic curves (work in progress with G. Hanrot on genus 2)
  - Ate pairing
  - pairings on Edwards curves

Concluding thoughts

- Characteristic 3 performs slightly better than characteristic 2
  - at least on our unified architecture
  - good overall performances vouch for stronger confidence in this observation
  - not true anymore on parallel architectures: the battle is not over!
- Unified operator
  - small but also competitively fast
  - parameter D to explore the area-time tradeoff
  - high scalability: support for larger extension degrees and higher levels of security
  - automatic VHDL generation: ultra-fast development
- Perspectives
  - parallel architectures (work in progress with N. Cortez-Duarte and N. Estibals)
  - hyperelliptic curves (work in progress with G. Hanrot on genus 2)
  - Ate pairing
  - pairings on Edwards curves
  - **AES-128-equivalent security!**

With thanks to our sponsor

Thank you for your attention

Questions?