CARDIS 2020

Smart Card Research and Advanced Application Conference

Secure and Efficient Delegation of Pairings with Online Inputs

Presenter: Matluba Khodjaeva

CUNY John Jay College of Criminal Justice
E-Mail: *mkhodjaeva@jjay.cuny.edu*

Co-authored by:
Giovanni Di Crescenzo, Perspecta Labs Inc. Basking Ridge, NJ, USA.
Delaram Kahrobaei, University of York. Heslington, York, UK.
Vladimir Shpilrain, City University of New York. New York, NY, USA.

11/18/2020

Computational Weaker Devices, Smart Cards

- Due to rapid growth in popularity of the Internet and wireless communications, many wireless E-commerce and business applications provide rapid and convenient resource accessing services to users.

Computational Weaker Devices, Smart Cards

- Due to rapid growth in popularity of the Internet and wireless communications, many wireless E-commerce and business applications provide rapid and convenient resource accessing services to users.
- As of 2015, 10.5 billion smart card IC chips are manufactured annually, including 5.44 billion SIM card IC chips.

Computational Weaker Devices, Smart Cards

- Due to rapid growth in popularity of the Internet and wireless communications, many wireless E-commerce and business applications provide rapid and convenient resource accessing services to users.
- As of 2015, 10.5 billion smart card IC chips are manufactured annually, including 5.44 billion SIM card IC chips.
- Considering the limited computing capability of smart cards or mobile devices, the security scheme design based on traditional public-key systems is a nontrivial challenge because most cryptographic algorithms require many expensive computations.

Computational Weaker Devices, Smart Cards

- Due to rapid growth in popularity of the Internet and wireless communications, many wireless E-commerce and business applications provide rapid and convenient resource accessing services to users.
- As of 2015, 10.5 billion smart card IC chips are manufactured annually, including 5.44 billion SIM card IC chips.
- Considering the limited computing capability of smart cards or mobile devices, the security scheme design based on traditional public-key systems is a nontrivial challenge because most cryptographic algorithms require many expensive computations.
- If public-key based cryptographic schemes are designed for smart cards, the computational cost on the user side is a critical issue for implementation because of their limited computing capability [T07].

Goal of Delegating Expensive Computations

- **Parties:** A computationally weaker client C and a computationally stronger server S

Goal of Delegating Expensive Computations

- **Parties:** A computationally weaker client C and a computationally stronger server S
- **Goal:** C has input x and need to compute $F(x)$ with help from S
  - F can be any function (e.g., a relatively expensive cryptographic computation)

Interaction Model and Requirements

- **Interaction model:**
  - *Offline phase* where C is not computationally limited (i.e., deployment of C's device)
  - *Online phase:* $C \to S$, $S \to C$

Interaction Model and Requirements

- **Interaction model:**
  - *Offline phase* where C is not computationally limited (i.e., deployment of C's device)
  - *Online phase:* $C \to S$, $S \to C$
- **Requirements:**
  - **Correctness:** At the end of a compliant execution of the protocol C outputs: $F(x)$
  - **Input Privacy:** Only minimal or no information about x should be revealed to S
  - **Output Security:** No S should force C's output $\neq F(x)$, except with very small probability
  - **Efficiency**
    - C's online runtime is $<<$ computing $F(x)$ without delegating computation
    - S's runtime is not $>>$ computing $F(x)$.

Our Delegation Problem: Computing a Pairing Function

- Let $\mathcal{G}_1$, $\mathcal{G}_2$ be additive cyclic groups of order l and $\mathcal{G}_T$ be a multiplicative cyclic group of the same order l, for some large prime l.

Our Delegation Problem: Computing a Pairing Function

- Let $\mathcal{G}_1$, $\mathcal{G}_2$ be additive cyclic groups of order l and $\mathcal{G}_T$ be a multiplicative cyclic group of the same order l, for some large prime l.
- A *bilinear map pairing* is a map $e : \mathcal{G}_1 \times \mathcal{G}_2 \to \mathcal{G}_T$ with the following properties:
  1. *Bilinearity:* for all $A \in \mathcal{G}_1$, $B \in \mathcal{G}_2$ and any $r, s \in \mathbb{Z}_l$, it holds that $e(rA, sB) = e(A, B)^{rs}$
  2. *Non-triviality:* if U is a generator for $\mathcal{G}_1$ and V is a generator for $\mathcal{G}_2$ then $e(U, V)$ is a generator for $\mathcal{G}_T$

Our Delegation Problem: Computing a Pairing Function

- Let $\mathcal{G}_1$, $\mathcal{G}_2$ be additive cyclic groups of order l and $\mathcal{G}_T$ be a multiplicative cyclic group of the same order l, for some large prime l.
- A *bilinear map pairing* is a map $e : \mathcal{G}_1 \times \mathcal{G}_2 \rightarrow \mathcal{G}_T$ with the following properties:
    1. *Bilinearity:* for all $A \in \mathcal{G}_1$, $B \in \mathcal{G}_2$ and any $r, s \in \mathbb{Z}_l$, it holds that $e(rA, sB) = e(A, B)^{rs}$
    2. *Non-triviality:* if U is a generator for $\mathcal{G}_1$ and V is a generator for $\mathcal{G}_2$ then $e(U, V)$ is a generator for $\mathcal{G}_T$
- Used as component in many cryptographic protocols
    - Cryptographic protocols based on **discrete logarithms** can usually be reformulated to work using pairings and result in space savings
    - **More capabilities:**
        - identity-based encryption [BF01],
        - short signatures [BLS01],
        - public-key encryption with keyword search [BDOP04],
        - 3-party key agreement [J00],
        - certificateless encryption and signatures [LAS07], etc.

In practical Curves, Operations Comparison in [BCN13]

Security level	Family-k	Pairing e	Scal. mul. in $\mathcal{G}_1$	Scal. mul. in $\mathcal{G}_2$	Exp. in $\mathcal{G}_T$
128-bits	BN-12	7.0	0.9	1.8	3.1
192-bits	BLS-12	47.2	4.4	10.9	17.5
	KSS-18	63.3	3.5	9.8	15.7
256-bits	BLS-24	115.0	5.2	27.6	47.1

In practical Curves, Operations Comparison in [BCN13]

- *Exponentiation operation* in $\mathcal{G}_T$ is more expensive than *scalar multiplication in* $\mathcal{G}_2$, and even more than *scalar multiplication* in $\mathcal{G}_1$

Security level	**Family-k**	**Pairing e**	**Scal. mul. in $\mathcal{G}_1$**	**Scal. mul. in $\mathcal{G}_2$**	**Exp. in $\mathcal{G}_T$**
128-bits	BN-12	7.0	0.9	1.8	3.1
192-bits	BLS-12	47.2	4.4	10.9	17.5
	KSS-18	63.3	3.5	9.8	15.7
256-bits	BLS-24	115.0	5.2	27.6	47.1

In practical Curves, Operations Comparison in [BCN13]

- *Exponentiation operation* in $\mathcal{G}_T$ is more expensive than *scalar multiplication in* $\mathcal{G}_2$, and even more than *scalar multiplication* in $\mathcal{G}_1$
- *Pairings* are almost 1 order of magnitude more expensive than *exponentiation* in $\mathcal{G}_T$

Security level	Family-k	Pairing e	Scal. mul. in $\mathcal{G}_1$	Scal. mul. in $\mathcal{G}_2$	Exp. in $\mathcal{G}_T$
128-bits	BN-12	7.0	0.9	1.8	3.1
192-bits	BLS-12	47.2	4.4	10.9	17.5
	KSS-18	63.3	3.5	9.8	15.7
256-bits	BLS-24	115.0	5.2	27.6	47.1

Previous Results

- *Girault et al. [ASIACRYPT05]:* Achieved input privacy, but no security

Previous Results

- *Girault et al. [ASIACRYPT05]:* Achieved input privacy, but no security
- *Guillevic et al. [CARDIS14]:* Improved efficiency, still no security

Previous Results

- *Girault et al. [ASIACRYPT05]:* Achieved input privacy, but no security
- *Guillevic et al. [CARDIS14]:* Improved efficiency, still no security
- *Chevallier-Mames et al. [CARDIS10] and Kang et al. [K05]:* satisfy result security, but not more efficient than non-delegated computation

Previous Results

- *Girault et al. [ASIACRYPT05]:* Achieved input privacy, but no security
- *Guillevic et al. [CARDIS14]:* Improved efficiency, still no security
- *Chevallier-Mames et al. [CARDIS10] and Kang et al. [K05]:* satisfy result security, but not more efficient than non-delegated computation
- *Canard et al. [ACNS14]:* 1st method with marginal efficiency than non-delegated computation for the KSS elliptic curve (but not the BN elliptic curve)

Previous Results

- *Girault et al. [ASIACRYPT05]:* Achieved input privacy, but no security
- *Guillevic et al. [CARDIS14]:* Improved efficiency, still no security
- *Chevallier-Mames et al. [CARDIS10] and Kang et al. [K05]:* satisfy result security, but not more efficient than non-delegated computation
- *Canard et al. [ACNS14]:* 1st method with marginal efficiency than non-delegated computation for the KSS elliptic curve (but not the BN elliptic curve)
- *Di Crescenzo et al. [ACNS20]:* 1st pairing delegation satisfying input privacy, security and efficiency with respect to all 4 most studied elliptic curves in several input cases (but not when case A, B are private online in the BN elliptic curve)

Our Contribution

- In this paper we show that when both inputs are only available in the *online phase*, bilinear-map pairings can be efficiently, privately and securely delegated to a single, possibly malicious, server.

Our Contribution

- In this paper we show that when both inputs are only available in the *online phase*, bilinear-map pairings can be efficiently, privately and securely delegated to a single, possibly malicious, server.
- Our results include 2 new protocols in the following cases both
  - A and B are *publicly* available
  - A and B are *privately* available.

Our Contribution

- In this paper we show that when both inputs are only available in the *online phase*, bilinear-map pairings can be efficiently, privately and securely delegated to a single, possibly malicious, server.
- Our results include 2 new protocols in the following cases both
  - A and B are *publicly* available
  - A and B are *privately* available.
- In both protocols improves the main performance metric (client's online runtime), with respect to all 4 most studied elliptic curves.
  - the client's online program only performs 1 exponentiation to a short (e.g., 128-bit) exponent in the most computationally intensive curve.

Our Contribution

- In this paper we show that when both inputs are only available in the *online phase*, bilinear-map pairings can be efficiently, privately and securely delegated to a single, possibly malicious, server.
- Our results include 2 new protocols in the following cases both
  - A and B are *publicly* available
  - A and B are *privately* available.
- In both protocols improves the main performance metric (client's online runtime), with respect to all 4 most studied elliptic curves.
  - the client's online program only performs 1 exponentiation to a short (e.g., 128-bit) exponent in the most computationally intensive curve.
- This improves over all previous protocols, where the client required either a larger number of exponentiations to short exponents or exponentiations to longer exponents, or more expensive pairing operations.

Our first protocol: A and B Public Online

Offline Input to C and S: $1^\sigma, 1^\lambda$, $desc(e)$

Offline phase instructions:

1. C randomly chooses $U \in \mathcal{G}_1$, $P \in \mathcal{G}_2$, $c \in \{1, \ldots, 2^\lambda\}$ and $r \in \mathbb{Z}_l^*$
2. C sets $\hat{r} = r^{-1} \mod l$, $Q_0 := \hat{r} \cdot P$, $v := e(U, P)$ and $ov = (c, r, U, P, Q_0, v)$

Online Inputs: $A \in \mathcal{G}_1$ and $B \in \mathcal{G}_2$ to both C and S, and ov to C

Online phase instructions:

1. C sets $Z := r(A - U)$, $Q_1 := c \cdot B + P$ and sends Z, Q_0, Q_1 to S
2. S computes $w_0 := e(A, B)$, $w_1 := e(A, Q_1)$, $w_2 := e(Z, Q_0)$
   S sends w_0, w_1, w_2 to C
3. (Membership Test:) C checks that $w_0, w_2 \in \mathcal{G}_T$
   (Probabilistic Test:) C checks that $w_1 = (w_0)^c \cdot w_2 \cdot v$
   (with this test, C implicitly checks that $w_1 \in \mathcal{G}_T$)
   If any of these tests fails, C **returns** $\perp$ and the protocol halts
   C **returns** $y = w_0$

Requirements of the 1st protocol

- *Correctness* holds: C obtains $y = w_0 = e(A, B)$ since A, B are known to S. We can show that Probabilistic and Membership Test always passed.

Requirements of the 1st protocol

- *Correctness* holds: C obtains $y = w_0 = e(A, B)$ since A, B are known to S. We can show that Probabilistic and Membership Test always passed.
- *Security* holds: main idea of the security is a *Probabilistic Test:* $e(A, Q_1) = e(A, B)^c \cdot e(Z, Q_0) \cdot e(U, P)$
  - c is a short (128 bits), random, online exponent
  - $P \in_R \mathcal{G}_2$, $U \in_R \mathcal{G}_1$, where $Q_0 = r^{-1} \cdot P$, $Q_1 = c \cdot B + P$, $Z = r(A - U)$
  - Result security follows by proving that
    - P random $\rightarrow Q_1$ does not leak c
    - If S sends incorrect (w_0', w_1', w_2'), it can only pass the probabilistic test with prob. $= 2^{-\lambda}$

Requirements of the 1st protocol

- *Correctness* holds: C obtains $y = w_0 = e(A, B)$ since A, B are known to S. We can show that Probabilistic and Membership Test always passed.
- *Security* holds: main idea of the security is a *Probabilistic Test:* $e(A, Q_1) = e(A, B)^c \cdot e(Z, Q_0) \cdot e(U, P)$
  - c is a short (128 bits), random, online exponent
  - $P \in_R \mathcal{G}_2$, $U \in_R \mathcal{G}_1$, where $Q_0 = r^{-1} \cdot P$, $Q_1 = c \cdot B + P$, $Z = r(A - U)$
  - Result security follows by proving that
    - P random $\rightarrow Q_1$ does not leak c
    - If S sends incorrect (w_0', w_1', w_2'), it can only pass the probabilistic test with prob. $= 2^{-\lambda}$
- *Efficiency comparison with other papers:*

Protocols	t_C	**Ratio:** t_C/t_F			
		BN-12 $\sigma = 461$	**BLS-12** $\sigma = 635$	**KSS-18** $\sigma = 508$	**BLS-24** $\sigma = 629$
[CARDIS10] §5.2	$e_T(\sigma) + m_1(\sigma) + m_2(\sigma)$	1.719	1.439	0.956	1.517
[ACNS14]§4.1	$e_T(\sigma) + m_1(\sigma)$	0.832	0.697	0.460	0.697
[ACNS20] §4.1	$2\,e_T(\lambda) + m_2(\lambda) + m_1(\sigma) + m_1(\lambda)$	0.485	0.310	0.235	0.272
This paper §3	$e_T(\lambda) + m_1(\sigma) + m_2(\lambda)$	0.326	0.216	0.158	0.179

A and B Private Online

- We investigate client-server protocols for secure pairing delegation, in the scenario where both of the pairing inputs are only known to the client in the *online phase*, and need to remain *private* from the server.

A and B Private Online

- We investigate client-server protocols for secure pairing delegation, in the scenario where both of the pairing inputs are only known to the client in the *online phase*, and need to remain *private* from the server.
- We presented 4 protocols in case when A, B are private online in this paper.

Most efficient protocol when A and B Private Online

Offline Input to C and S: $1^\sigma, 1^\lambda$, *desc*(e)

Offline phase instructions:

1. C randomly chooses $U_0, U_1 \in \mathcal{G}_1$, $P_0, P_1 \in \mathcal{G}_2$, $c \in \{1, \ldots, 2^\lambda\}$, $r_0, r_1, r_2 \in \mathbb{Z}_l^*$
2. C sets
   - $v_i := e(U_i, P_i)$, $Q_i := \hat{r}_i \cdot P_i$ where $\hat{r}_i = r_i^{-1} \mod l$, for $i = 0, 1$
   - $\hat{r}_2 := r_2^{-1}$, $Q_{2,1} = -r_2 \cdot P_0$ and $Q_{3,1} = r_2 \cdot P_1$
3. C sets $ov = (c, r_0, r_1, r_2, \hat{r}_2, U_0, U_1, P_0, P_1, Q_0, Q_1, Q_{2,1}, Q_{3,1}, v_0, v_1)$

Online Input to C: $A \in \mathcal{G}_1$, $B \in \mathcal{G}_2$, and ov

Online phase instructions:

1. C sets
   - $Z_0 := r_0(A - U_0)$, $Z_1 := r_1(A - U_1)$, $Z_2 := \hat{r}_2 \cdot A$ and
   - $Q_{2,0} = Q_{3,0} := r_2 \cdot B$, $Q_2 := Q_{2,0} + Q_{2,1}$, $Q_3 := c \cdot Q_{3,0} + Q_{3,1}$

   C sends $Z_0, Z_1, Z_2, Q_0, Q_1, Q_2, Q_3$ to S
2. S computes

   $$w_0 := e(Z_0, Q_0),\ w_1 := e(Z_1, Q_1),\ w_2 := e(Z_2, Q_2),\ w_3 := e(Z_2, Q_3)$$

   S sends w_0, w_1, w_2, w_3 to C
3. (Membership Test:) C checks that $w_0, w_1, w_2 \in \mathcal{G}_T$

   C computes $y = w_0 \cdot w_2 \cdot v_0$

   (Probabilistic Test:) C checks that $w_3 = (y)^c \cdot w_1 \cdot v_1$

   (with this test, C implicitly checks that $w_3 \in \mathcal{G}_T$)

   If any of these tests fails, C **returns** $\perp$ and the protocol halts

   C **returns** y

Requirements of the Second Protocol

- *Correctness* holds:

$$
\begin{aligned}
y &= w_0 \cdot w_2 \cdot v_0 = e(Z_0, Q_0) \cdot e(Z_2, Q_2) \cdot e(U_0, P_0) \\
&= e(r_0(A - U_0), r_0^{-1} P_0) \cdot e(r_2^{-1} A, r_2(B - P_0)) \cdot e(U_0, P_0) \\
&= e(A - U_0, P_0) \cdot e(A, B - P_0) \cdot e(U_0, P_0) \\
&= e(A, P_0) \cdot e(U_0, P_0)^{-1} \cdot e(A, B) \cdot e(A, P_0)^{-1} \cdot e(U_0, P_0) \;=\; e(A, B).
\end{aligned}
$$

We can show that Probabilistic and Membership Test always passed.

Requirements of the Second Protocol

- *Correctness* holds:

$$
\begin{aligned}
y &= w_0 \cdot w_2 \cdot v_0 = e(Z_0, Q_0) \cdot e(Z_2, Q_2) \cdot e(U_0, P_0) \\
&= e(r_0(A - U_0), r_0^{-1} P_0) \cdot e(r_2^{-1} A, r_2(B - P_0)) \cdot e(U_0, P_0) \\
&= e(A - U_0, P_0) \cdot e(A, B - P_0) \cdot e(U_0, P_0) \\
&= e(A, P_0) \cdot e(U_0, P_0)^{-1} \cdot e(A, B) \cdot e(A, P_0)^{-1} \cdot e(U_0, P_0) \; = \; e(A, B).
\end{aligned}
$$

  We can show that Probabilistic and Membership Test always passed.
- The *privacy* property of the protocol against any malicious S follows by observing that C's message $(Z_0, Z_1, Z_2, Q_0, Q_1, Q_2, Q_3)$ to S does not leak any information about C's inputs A, B.

Requirements of the Second Protocol

- *Correctness* holds:

$$
\begin{aligned}
y = w_0 \cdot w_2 \cdot v_0 &= e(Z_0, Q_0) \cdot e(Z_2, Q_2) \cdot e(U_0, P_0) \\
&= e(r_0(A - U_0), r_0^{-1} P_0) \cdot e(r_2^{-1} A, r_2(B - P_0)) \cdot e(U_0, P_0) \\
&= e(A - U_0, P_0) \cdot e(A, B - P_0) \cdot e(U_0, P_0) \\
&= e(A, P_0) \cdot e(U_0, P_0)^{-1} \cdot e(A, B) \cdot e(A, P_0)^{-1} \cdot e(U_0, P_0) \ = \ e(A, B).
\end{aligned}
$$

  We can show that Probabilistic and Membership Test always passed.
- The *privacy* property of the protocol against any malicious S follows by observing that C's message $(Z_0, Z_1, Z_2, Q_0, Q_1, Q_2, Q_3)$ to S does not leak any information about C's inputs A, B.
- *Security* holds: main idea of the security is a *Probabilistic Test:*

$$e(Z_2, Q_3) = y^c \cdot e(Z_1, Q_1) \cdot e(U_1, P_1)$$

  We showed in the paper, if S sends incorrect (w_0', w_1', w_2', w_3'), it can only pass the probabilistic test with prob. $= 2^{-\lambda}$

Efficiency comparison with other papers

Protocols	t_C	**Ratio:** t_C/t_F			
		BN-12 $\sigma = 461$	**BLS-12** $\sigma = 635$	**KSS-18** $\sigma = 508$	**BLS-24** $\sigma = 629$
[CARDIS10] §4.1	$5\,e_T(\sigma) + m_2(\sigma)$	2.606	2.182	1.453	2.337
[K05] §3	$3\,e_T(\sigma) + m_2(\sigma) + m_1(\sigma)$	1.719	1.439	0.956	1.517
[CARDIS14] §5.1	$2\,e_T(\sigma) + 2\,m_2(\sigma) + 2\,m_1(\sigma)$	1.658	1.391	0.917	1.390
[ACNS20] Π_1	$3\,e_T(\lambda) + m_2(\sigma) + m_2(\lambda)$ $+3\,m_1(\sigma) + 2\,m_1(\lambda)$	1.161	0.823	0.578	0.697
This paper: Π_0	$e_T(\sigma) + e_T(\lambda) + m_2(\sigma)$ $+m_2(\lambda) + 2\,m_1(\sigma)$	1.155	0.911	0.617	0.874
This paper: Π_2	$3\,e_T(\lambda) + m_2(\sigma) + 2\,m_2(\lambda)$ $+2\,m_1(\sigma) + m_1(\lambda)$	1.072	0.760	0.550	0.694
This paper: Π_3	$2\,e_T(\lambda) + m_2(\sigma) + 2\,m_2(\lambda)$ $+1\,m_1(\sigma) + m_1(\lambda)$	1.002	0.729	0.502	0.604
This paper §4	$e_T(\lambda) + m_2(\sigma)$ $+m_2(\lambda) + 3\,m_1(\sigma)$	0.843	0.635	0.425	0.511

Conclusions

- In this paper we showed techniques for a computationally weaker client (e.g. smartcards) efficiently, privately and securely delegate pairings to a single, possibly malicious, server, in the input scenario where both inputs are not available until the online phase.

Conclusions

- In this paper we showed techniques for a computationally weaker client (e.g. smartcards) efficiently, privately and securely delegate pairings to a single, possibly malicious, server, in the input scenario where both inputs are not available until the online phase.
- We proposed new protocols in the scenario where
  1. both inputs A, B are publicly available;
  2. both inputs A, B are known to C but should remain private from S.

Conclusions

- In this paper we showed techniques for a computationally weaker client (e.g. smartcards) efficiently, privately and securely delegate pairings to a single, possibly malicious, server, in the input scenario where both inputs are not available until the online phase.
- We proposed new protocols in the scenario where
  1. both inputs A, B are publicly available;
  2. both inputs A, B are known to C but should remain private from S.
- For the first time in the state of art when A, B are private, we showed the C's online runtime with respect to non-delegated computation for all 4 practical curves including BN curve.

Conclusions

- In this paper we showed techniques for a computationally weaker client (e.g. smartcards) efficiently, privately and securely delegate pairings to a single, possibly malicious, server, in the input scenario where both inputs are not available until the online phase.
- We proposed new protocols in the scenario where
  1. both inputs A, B are publicly available;
  2. both inputs A, B are known to C but should remain private from S.
- For the first time in the state of art when A, B are private, we showed the C's online runtime with respect to non-delegated computation for all 4 practical curves including BN curve.
- In both protocols *efficiency gains* obtained by our resulting protocols with respect to the main metric (client's online runtime).

Conclusions

- In this paper we showed techniques for a computationally weaker client (e.g. smartcards) efficiently, privately and securely delegate pairings to a single, possibly malicious, server, in the input scenario where both inputs are not available until the online phase.
- We proposed new protocols in the scenario where
  1. both inputs A, B are publicly available;
  2. both inputs A, B are known to C but should remain private from S.
- For the first time in the state of art when A, B are private, we showed the C's online runtime with respect to non-delegated computation for all 4 practical curves including BN curve.
- In both protocols *efficiency gains* obtained by our resulting protocols with respect to the main metric (client's online runtime).
- Our techniques improve the state of the art on both scenarios.

References

[BCN13] J.W. Bos, C. Costello, M. Naehrig, *Exponentiating in pairing groups.* In: Lange T., Lauter K., Lisoněk P. (eds) SAC 2013. LNCS vol 8282. Springer.

[BF01] D. Boneh, M. Franklin, *Identity-based Encryption from the Weil Pairing*. In: Proc. of CRYPTO 2001. LNCS vol. 2139, Springer.

[BDOP04] D. Boneh, G. Di Crescenzo, R. Ostrovsky, G. Persiano, *Public Key Encryption with Keyword Search*. In: Proc. of EUROCRYPT 2004. LNCS vol. 3027, Springer.

[BLS01] D. Boneh, B. Lynn, H. Shacham, *Short Signatures from the Weil Pairing.* In: Boyd C. (eds) Advances in Cryptology — ASIACRYPT 2001. LNCS vol 2248. Springer.

[ACNS14] S. Canard, J. Devigne, O. Sanders, *Delegating a pairing can be both secure and efficient.* In: Boureanu I., Owesarski P., Vaudenay S. (eds) Applied Cryptography and Network Security. ACNS 2014. LNCS vol 8479. Springer.

[CARDIS10] B. Chevallier-Mames, J.S. Coron, N. McCullagh, D. Naccache, M. Scott, *Secure delegation of elliptic-curve pairing. Cryptology ePrint Archive.* In: Proc. of CARDIS 2010. LNCS vol 6035. Springer. Also IACR EPrint 2005/150.

References

[ACNS20] G. Di Crescenzo, M. Khodjaeva, D. Kahrobaei, V. Shpilrain, *Secure and Efficient Delegation of Elliptic-Curve Pairing*. In: Proc. of ACNS 2020. LNCS, vol 12146. Springer, Cham.

[ASIACRYPT05] M. Girault, D. Lefranc, *Server-aided verification: Theory and practice.* In: Roy, B.K. (ed.) ASIACRYPT 2005. LNCS, vol. 3788, pp. 605–623.

[CARDIS14] Guillevic, A., Vergnaud, D., *Algorithms for outsourcing pairing computation.* In: Proc. of CARDIS 2014, LNCS vol. 8968. Springer.

[J00] Antoine Joux, *A One Round Protocol for Tripartite Diffie-Hellman.* In: Proc. of ANTS 2000, pp. 385-394.

[K05] B.G. Kang, M.S. Lee, J.H. Park, *Efficient delegation of pairing computation.* In: IACR Cryptology ePrint Archive, n. 259, 2005.

[LAS07] J.K. Liu, M.H. Au, W. Susilo, *Self-generated-certificate publickey cryptography and certificateless signature/encryption schemein the standard model.* In: Proc. ACM Symp. on Information, Computer and Communications Security. ACM Press (2007).

[T07] Tseng, Y.M., *A resource-constrained group key agreement protocol for imbalanced wireless networks.* In: Computers Security,26(4), 331–337, 2007

Thank You!

Questions?