Forging a winning bid in Mu-Varadharajan’s E-auction

Hung-Yu Chien
Department of Information Management
National Chi Nan University
TAIWAN, R.O.C.
redfish6@ms45.hinet.net

Abstract: - Mu and Varadharajan proposed an important sealed bid e-auction scheme that protects even the winner’s real identity and secures the fair exchange of electronic goods and electronic payment. This paper presents the forgery attack on the Mu-Varaharajan’s scheme: an attacker can easily forge valid bids and win the auction without any payment. The improvements to conquer the weaknesses are then presented.

Key-Words: - Cryptography, auction, verifiable encryption, knowledge proof.

1 Introduction

As Internet becomes an indispensable tool of our daily life, the research on e-auction schemes, especially the sealed-bid schemes like [16, 17], derives more and more attention in recent years. A sealed-bid auction is one that the secret bids are protected from disclosure before the bidding deadline. The common requirements of sealed-bid auctions are: (1) secrecy of the bids before the deadline, (2) fairness of the process, (3) non-repudiation of submitted bids, and (4) validity of the outcome. Franklin and Reiter [1] utilized the verifiable secret sharing [2] to propose a sealed-bid auction in which the bids are open after the bidding is closed. This type of auction is typically used in auctioning of artwork, real estate, and government contracts. The auction service is distributed over multiple auction servers. The bidders have to trust that the number of colluded servers would not be greater than the threshold value. This requirement may be unrealistic in practice and the result is not public verifiable. In some applications, even the privacy of losing bids should be protected. Kikuchi et al.’s schemes [7] reveal no auction bids except the winning bid. But, the schemes also require multiple servers, and assume that most servers would not collude. Moreover, there is no guarantee that the winner will pay. Zhang et al.’s scheme [5] requires the auctioneer and each bidder interactively run the two millionaires protocol [6] to decide whether the bidder bids at a specified price. The process is run from the highest price down to the winner’s price. This process is very costly. To provide an efficient auction scheme, Kobayashi et al. [8] designed a new scheme in which most of the computations involve only hashing operations. The scheme greatly improves the computational efficiency. But, it demands lots of interactions between the auctioneer and the bidders in the bid-opening phase. The interaction cost is high. The Watanabe-Imai scheme [10], using verifiable encryption [9], has several merits: (1) there is no requirement of multiple servers to protect security, (2) bidders are not involved in the bid opening phase, and (3) the trusted third party is involved only in optimistic sense [10]. However, the anonymity of bidders is not provided.

Mu and Varadaharajan [14] proposed a new sealed-bid auction scheme that provides the maximum bidder privacy in which only the anonymous identity of the winner is revealed when the bidding is closed. This is useful when protecting the winner from coercion. In addition, the scheme expected to own several practical merits: (1) no requirement of multiple servers is needed to ensure fairness and privacy, (2) the winner’s payment is ensured, and (3) the exchange of bidder’s information and seller’s digital goods is fairly implemented. However, it will be shown that an attacker can easily forge valid bids and get the goods. Even though the auctioneer and the trusted party find the cheating finally, they cannot identify the cheater and cannot resume the goods. The rest of this article is organized as follows. Section 2 first introduces some cryptographic preliminaries, and then reviews the Mu-Varadaharajan scheme. Section 3 demonstrates the attack. Section 4 states our conclusions.

2 Review of Mu-Varadaharajan’s Scheme

Mu and Varadaharajan proposed their e-auction scheme in which the fairness of the process and the privacy of bids are protected without requiring multiple servers. The scheme is based on verifiable encryption [13], optimistic fair exchange [11], knowledge proof [12, 15] and the blind Nyberg-Rueppel digital signature. The scheme also ensures the winner's payment, and fair exchange of bidder's information and the digital goods. The Mu-Varadaharajan scheme is somewhat complicated. To present the scheme clearly, the cryptographic primitives- verifiable encryption, optimistic fair exchange, knowledge proofs and the blind Nyberg-Rueppel digital signature- are first introduced below.

2.1 The cryptographic primitives

The verifiable encryption, optimistic fair exchange, knowledge proof and blind Nyberg-Rueppel digital signature are introduced here in functional point of view. They are denoted as simple notations here to make the presentation of the Mu-Varadaharajan scheme clear.

$VE(m)$: The verifiable encryption of message m. Verifiable encryption consists of a ciphertext under the trusted authority's (*TA*) public key and a non-interactive proof that the plaintext corresponding to the ciphertext is indeed the required information [13].

$OFE(a,b)$: The optimistic fair exchange of a and b. Fair exchange is a protocol in which two un-trusted parties run the protocol to exchange their information fairly [11, 13]. After the protocol, either both parties get the other's information or neither party gets anything. An optimistic fair exchange protocol assumes the existence of a trusted authority (*TA*) who involves the protocol in an optimistic manner. That is, *TA* will not involve the protocol directly, but becomes apparent only when resolving a dispute later. An optimistic fair exchange protocol makes use of verifiable encryption from which *TA* can decrypt the encryption when necessary.

$BlindSig\_DL(m,Y)$: The blind signature of message m, where Y is the signer's public key and the signature is based on the Discrete Logarithm problem (DLP). With a blind signature scheme, even the signer cannot trace the signature back to its corresponding signing instance and cannot link any two blind signatures. In Camenisch-Piveteau-Stalder's blind signature scheme [12], $BlindSig\_DL(m,Y)$ = (r,s) and the verification equation is $m = g^{-s}Y^r r \mod p$, where g is the primitive element in $GF(p)$.

$DLP(x : g^x)$: The proof of knowledge of discrete logarithm of $g^x \bmod p$. Knowledge proof is a cryptographic primitive that proves the knowledge of some secret without revealing anything about the secret [12, 15]. We denote $EQ\_DLP(x : \log_g y_1 = \log_h y_2)$ to the proof of the knowledge $x = \log_g y_1 = \log_h y_2$, where $y_1 \equiv_p g^x$, $y_2 \equiv_p h^x$, and g and h are public bases. Denote $EQ\_DDLP(\varepsilon : \log_{h'} \overline{h'} = \log_g(\log_\alpha \alpha'))$ to the proof of the knowledge $\varepsilon = \log_{h'} \overline{h'} = \log_g(\log_\alpha \alpha')$, where $\overline{h'} \equiv_q h'^{\varepsilon}$, $\alpha' \equiv_p \alpha^k$ and $k \equiv_q g^{\varepsilon}$. Denote $EQ\_REP[(\varepsilon,1) : \overline{h'} \equiv_q h'^{\varepsilon}, R_1 \equiv_q r_1 g^{\varepsilon}]$ to the proof of the knowledge ε such that $\varepsilon = \log_{h'} \overline{h'}$ and $\varepsilon =$ the discrete logarithm of the g-part representation of R_1 to bases g and r_1. The readers may refer to [12, 15] for the detailed implementations.

2.2 The Mu-Varadaharajan e-auction in functional point of view

The Mu-Varadaharajan protocol is introduced first in functional point of view to help the presentation of the detailed protocol clear later. The scheme involves of 3 roles: an auction server S, a financial institution F, and several bidders $\{B_i\}$. The financial institution F assumes the role of trusted third party. The scheme consists of three phases: the anonymous account and anonymous bid certificate generation phase, the bid casting phase, and the bid opening and fair exchange phase.

The anonymous account and anonymous bid certificate generation phase

In this phase, each bidder B_i first applies for an anonymous account A_i from F. The bidder decides his bid u and his bid commitment v that is a deterministic function of u. The bid u is a concatenation of the bid value and a secret random value. Based on this account A_i, B_i requests a certificate $Cert_{B_i}$ for his bid commitment v from F, using the blind digital signature scheme. Therefore, the auctioneer S and the financial institution F cannot trace the bidder when the bidder submits the bid $Cert_{B_i}$ later.

The bid-casting phase

When B_i wants to submit his bid, he first gets a unique bidding number N from S and then submits his bid-related data to S. The bid-related data consists of the bid itself, an escrowed bid opener, and some knowledge proofs of the correctness of the bid opener. The bid itself and the escrowed bid opener also serve as knowledge proofs of the bid certificate $Cert_{B_i}$. The escrowed bid opener is also a verifiable encryption from which the trusted authority (*TA*) can decrypt it to get the bid u and the bid commitment v if there is a dispute later.

The bid opening and fair exchange phase

After the bid-casting phase, all bidders anonymously submit their bids u along with the corresponding bidding numbers to S. From these data, S broadcasts the highest bid. The winners who cast the highest bid run the fair exchange protocol with S to interchange the digital goods and a new knowledge proof of the bids. From the bid itself and this new proof, S acquires the bid u and submits it to F. Based on u, F identifies the anonymous account and transfers the money.

2.3 The Mu-Varadaharajan protocol

The system parameters and the notation are introduced first, and then the Mu-Varadaharajan scheme is presented in detail. $p = 2q+1$ is a prime number, where $q = p_1 p_2$ and both p_1 and p_2 are primes. Z_p^* is a primitive group of order $p-1$. Z_q^* is a primitive group of order $\phi(q)$. $g \in Z_p^*$ is a generator of order q, and $g \in Z_q^*$. $h()$ is a secure one-way function. Let x be F 's secret key and $h \equiv_p g^x$ and $h' \equiv_q g^x$ be its public keys. F also chooses two random numbers w_1 and w_2, computes $g_1 \equiv_p g^{w_1}$, $g_2 \equiv_p g^{w_2}$, $h_1 \equiv_p {g_1}^x$, and $h_2 \equiv_p {g_2}^x$. F makes ($p, q, g, h, h', g_1, g_2, h_1, h_2$) public. Now each phase is introduced in detail as follows.

The anonymous account and anonymous bid certificate generation phase

B_i chooses a random number σ_i, and computes his alias $A_i \equiv_p g^{\sigma_i}$. He uses the alias A_i to apply for an anonymous account from F. He also deposits enough money in that account. He then decides his bid value, and lets u be the concatenation of his bid value and a secret random value. He computes the bid commitment $v \equiv_p g_1 {g_2}^u$. The bid commitment v associated with the alias A_i is registered in the financial institution F, and B_i is given a certificate $w \equiv_p v^x$ along with the proof $\log_v w = \log_g g^x$. Now B_i applies the blind Nyberg-Rueppel digital signature scheme to acquire an anonymous account certificate $Cert_{B_i}$ for the bid u as follows. $y \in_R Z_q$ denotes that y is a number randomly chosen from Z_q.

1. B_i sends the proof $DLP(\sigma_i : A_i)$ to authenticate himself to F. After verifying the proof, F randomly chooses a number t, computes $\delta \equiv_p v^t$ and forwards it to B_i.

2. B_i chooses three random numbers (y, x_1, x_2), computes $\alpha \equiv_p w^y$, $\beta \equiv_p v^y$ and $\lambda \equiv_p {h_1}^{x_1} {h_2}^{x_2}$. He forms the message $m = h(\alpha, \beta, \lambda)$, chooses random numbers (a, b), calculates $r \equiv_p m\beta^a \delta^{by}$ and sends $m' \equiv_q r / b$ to F.

3. F signs on the blinded message m' by computing $s' \equiv_q m'x + t$. He sends s' to B_i.

4. B_i computes $s \equiv s'b + a$. Now the anonymous account certificate for the bid u is defined as $Cert_{B_i} \overset{def}{=} \{ \alpha, \beta, \lambda, r, s \}$. The verification equation is as follows.

$$h(\alpha,\beta,\lambda) \overset{?}{=} \beta^{-s}\alpha^{r} r \bmod p \qquad (1)$$

Figure 1. Requesting the anonymous certificate (the end of the paper)

The bid-casting phase

To submit his bid, B_i first gets a unique bidding number N from S and then submits his bid-related data to S. The bid-related data consists of the bid itself, an escrowed bid opener, and the knowledge proof of the correctness of the bid opener. The bid-casting protocol is as followed.

1. S generates a random challenge $c = h(S \| date \| time \| ...)$ which is distinct for each bid. S sends c to B_i.

2. Upon receiving c, B_i computes $r_1 \equiv_q x_1 + cy$ and $r_2 \equiv_q x_2 + ucy$, using his bid u. He sends ($N, r_1, r_2, Cert_{B_i}$) to S.

3. S verifies the validity of the certificate and the values (r_1, r_2) by checking whether the equations $h(\alpha,\beta,\lambda) \overset{?}{=} \beta^{-s}\alpha^{r} r \bmod p$ and $h_1^{r_1} h_2^{r_2} \overset{?}{=} \alpha^{c}\lambda \bmod p$ hold. If so, he stores ($c, r_1, r_2, Cert_{B_i}$) and sends a challenge c' to B_i for computing the escrowed bid opener.

4. B_i computes the bid opener $\{r'_1, r'_2\}$, which is then encrypted using the trusted third party's public key h. The computed values include $k, \overline{h'}, R_1, R_2, \alpha', \text{and } \lambda'$. B_i sends { $\overline{h'}, R_1, R_2, \alpha', \lambda'$ } to S. In addition, B_i sends S the knowledge proofs $EQ\_DDLP(\varepsilon : \log_{h'} \overline{h'} = \log_g(\log_\alpha \alpha'))$, $EQ\_DDLP(\varepsilon : \log_{h'} \overline{h'} = \log_g(\log_\lambda \lambda'))$, $EQ\_REP[(\varepsilon,1) : \overline{h'} \equiv_q h'^{\varepsilon}, R_1 \equiv_q r_1 g^{\varepsilon}]$, and $EQ\_REP[(\varepsilon,1) : \overline{h'} \equiv_q h'^{\varepsilon}, R_2 \equiv_q r_2 g^{\varepsilon}]$.

5. S verifies the validity of the bid opener by checking whether the following equation holds. He also verifies the knowledge proofs.

$$h_1^{R_1} h_2^{R_2} \overset{?}{=} \alpha'^{c'} \lambda' \bmod p \qquad (2)$$

Figure 2. bid-casting protocol (the end of the paper)

The bid opening and fair exchange phase

After the bid-casting phase, all bidders anonymously submit their bids u along with the corresponding bidding numbers to S. From these data, S broadcasts the highest bid and a new challenge c''. The winner who casts the highest bid computes $r''_1 \equiv_q x_1 + c''y$ and $r''_2 \equiv_q x_2 + uc''y$. The winner and S then run a fair exchange protocol $OFE(r'', goods)$, where $r'' = (N, r''_1, r''_2)$ and $goods$ is the digital goods. From r'' and (r_1, r_2), S derives y and u from Equation (3), and checks whether u contains the highest bid value. If so, S sends $v \equiv_p g_1 g_2^{u}$ to F , who can find the match of this value and the one in his alias list. With such a match, F transfers the money from the anonymous account A_i to S's account. If the winner defaults to send r'', S sends ($Cert_{B_i}$, c', R_1, R_2) to the trusted third party F , who then decrypts (R_1, R_2) and sends (r'_1, r'_2) to S. From (c', r'_1, r'_2) and (c, r_1, r_2), S can derive the values u and v.

$$\begin{aligned} \frac{r_1 - r''_1}{c - c''} &= \frac{x_1 + cy - x_1 - c''y}{c - c''} = y \\ \frac{r_2 - r''_2}{c - c''} &= \frac{x_2 + cyu - x_2 - uc''y}{c - c''} = yu \end{aligned} \qquad (3)$$

3 Forgery Attack and The Improvements

3.1 Forging a winning bid

The forgery attack will be shown here. An attacker who has no account in the financial institution can forge valid bid token and certificate to cheat the auctioneer to accept his bid. Let A be the attacker who does not follow the protocol to apply for an account in the financial institution. A first decides the highest bid u, and forges the anonymous account certificate $Cert_A$ for the bid u.

A then participates the bid-casting protocol and the bid opening and fair exchange phase to win the auction. Even though the financial institution and the auctioneer finally detect the cheating, they cannot identify the attacker and cannot resume the goods. The details are described as follows.

A chooses the highest bid u. A chooses 5 random numbers k_1, k_2, x_1, x_2 and y. A computes $\alpha \equiv_p (h_1 h_2{}^u)^y$, $\beta \equiv_p \alpha^{k_1}$, $\lambda \equiv_p h_1{}^{x_1} h_2{}^{x_2}$, $m = h(\alpha, \beta, \lambda)$, $r \equiv_p m \cdot \alpha^{k_2}$, and $s \equiv_q k_1{}^{-1}(r + k_2)$. Let $Cert_A$ = { $\alpha, \beta, \lambda, r, s$ }. $Cert_A$ is a valid certificate, and it satisfies the verification equation (1) as follows. $\beta^{-s} \alpha^r r \equiv_p \alpha^{k_1 \cdot k_1{}^{-1} \cdot (-r-k_2)} \cdot \alpha^r \cdot m\alpha^{k_2} \equiv_{\mathrm{p}} m \equiv_p h(\alpha, \beta, \lambda)$.

So $Cert_A$ is valid. Now A participates the bid-casting protocol using this certificate as follows. Upon receiving c and N, A prepares his bid by computing $r_1 \equiv_q x_1 + cy$ and $r_2 \equiv_q x_2 + ucy$, and sends $(N, r_1, r_2, Cert_A)$ to S. S will accept this bid because $Cert_A$ satisfies Equation (1) and (r_1, r_2) satisfies $h_1{}^{r_1} h_2{}^{r_2} \overset{?}{=} \alpha^c \lambda$. After the verification, S forwards c' to A for computing the escrowed bid opener. A now prepares the bid opener as follows. A chooses a random number ε, and computes $r'_1 \equiv_q x_1 + c'y, r'_2 \equiv_q x_2 + uc'y$, $k \equiv_q g^\varepsilon$, $\overline{h'} \equiv_q h'^\varepsilon, R_1 \equiv_q r'_1 k, R_2 \equiv_q r'_2 k$, $\alpha' \equiv_p \alpha^k$, and $\lambda' \equiv_p \lambda^k$. A then sends { $\overline{h'}, R_1, R_2, \alpha', \lambda'$ } to S. It is easy to check that the data satisfies Equation (2). It is also obvious that A can prepares the proofs $EQ\_DDLP(\varepsilon : \log_{h'} \overline{h'} = \log_g(\log_\alpha \alpha'))$, $EQ\_DDLP(\varepsilon : \log_{h'} \overline{h'} = \log_g(\log_\lambda \lambda'))$, $EQ\_REP[(\varepsilon,1) : \overline{h'} \equiv_q h'^\varepsilon, R_1 \equiv_q r_1 g^\varepsilon]$, and $EQ\_REP[(\varepsilon,1) : \overline{h'} \equiv_q h'^\varepsilon, R_2 \equiv_q r_2 g^\varepsilon]$ because A knows the secret ε and has computed the data { $\overline{h'}, R_1, R_2, \alpha', \lambda'$ } using this ε. Now S accepts A's submitted bid and bid opener.

After the bid-casting phase, A submits his bid u and N to S. After all bidders have submitted their bids, S announces the highest bid and a new challenge c''. Since A casts the highest bid, he prepares $r''_1 \equiv_q x_1 + c''y$ and $r''_2 \equiv_q x_2 + uc''y$, and lets $r'' = (N, r''_1, r''_2)$. A and S run $OFE(r'', goods)$, where $goods$ is the seller's digital goods. After running the fair exchange protocol, A gets the digital goods, and S derives the u from the r''. However, the financial institution cannot transfer any money because he cannot find the account corresponding to $v \equiv_p g_1 g_2{}^u$. The attacker succeeds in cheating the auctioneer and obtaining the goods.

3.2 The improved scheme

The key weakness of Mu-Varadaharajan's protocol is that an attacker can forge valid certificates and use this certificate to participate the auction process. And, the weakness results from wrong design of the certificate generation phase in which the public key in the verification equation (Equation 1) of the blind signature is wrongly replaced with a user chosen value β. To improve the weakness, any secure blind signature scheme to apply for a certificate for the data ($\alpha \equiv_p w^y, \lambda \equiv_p h_1{}^{x_1} h_2{}^{x_2}$) is sufficient, and a blind signature scheme with message recovery (like Camenisch-Piveteau-Stalder's blind signature scheme used in Mu-Varadaharajan's protocol) is not necessary, because we do not require the message recovery property. However, to keep the presentation consistent, we, based on Camenisch-Piveteau-Stalder's blind signature, show one improvement in Figure 3. The improved verification equation for the certificate is depicted in Equation (4), and the bid casting phase and the bid opening and fair exchange phase remain the same, except that the certificate is updated as $Cert_{B_i} \overset{def}{=}$ { α, λ, r, s } and the verification equation is updated as Equation (4).

Figure 3. Improved certificate generation phase (the end of the paper)

4 Conclusion

This paper has demonstrated the forgery attack on Mu-Varadharajan' e-auction scheme. An attacker can easily forge valid bids that pass all the verification processes, and then win the goods. We also have proposed our improvements to conquer the weaknesses.

Acknowledgements: This research is partially supported with project number NSC 95-2221-E-324-008-MY2.

References:

[1] Franklin, M. K., and M.K. Reiter (1996). The design and implementation of a secure auction service. IEEE trans. Software Engineering, 22(5), 302-312.

[2] Feldman, P (1987). A practical scheme for non-interactive verifiable secret sharing. In: Proceedings of 28th IEEE Symp. Foundations of Computer Science, 427-437.

[3] Kikuchi, H., M. Harkary, and J.-D. Tyger (1998). Multi-round anonymous auction protocol. In: Proc. of the first IEEE workshop on Dependable and Real-Time E-Commerce System,62-69.

[4] Shamir, A (1979). How to share a secret, Comm. ACM, 22(11), 612-613.

[5] Zhang, F., Q. Li, and Y. Wang (2000). A new secure electronic auction scheme. In: Eurocomm2000, Information Systems for Enhanced Public Safety and Security,54-56.

[6] Yao, A. C. (1986). Protocol for secure computations. In: Proceedings of the 27th IEEE Symposium on Foundations of Computer Science, 162-167.

[7] Kikuchi, H., S. Hotta, and K. Abe (2000). Distributed auction services resolving winner and winning bid without revealing privacy of bids. In: IEEE Seventh International Conference on Parallel and Distributed Systems, 307-312.

[8] Kobayashi, K., H. Morita, K. Suzuki, and M. Hakuta (2001). Efficient sealed-bid auction by using one-way functions. IEICE trans. Fund., E84-A(1), 289-294.

[9] Nachache, D., J. Stern (1998). A new public key cryptosystem based on higher residues. In: Proc. of ACM 5th Conference on Computer & Communications Security, 59-66.

[10] Watanabe, Y., and H. Imai (2000). Reducing the round complexity of a sealed-bid auction protocol with an off-line TTP. In: ACM 2000, CCS'00, Athens, Greece, 80-86.

[11] Asokan, N., V. Shoup, and M. Waidner (1998). Asynchronous protocols for optimistic fair exchange. In: Advances in Cryptography-Eurocrypt'98, LNCS 1403, Springer-Verlag, 5091-606.

[12] Camenisch, J., J. Piveteau, M. Stadler (1994). Blind signatures based on the discrete logarithm problem. In: Advances of Cryptography- Eurocrypt'94, Springer-Verlag, 428-432.

[13] Ateniese, G. (1999). Efficient verifiable encryption (and fair exchange) of digital signatures. In: Proc. Of ACM CCS'99, 138-146.

[14] Mu, Y., and V. Varadharajan (2000). An internet anonymous auction scheme. In: Proceedings of information Security and Cryptography 2000-ICISC2000, LNCS, Springer-Verlag, 171-182.

[15] Camenisch, J., and M. Stadler (1997). Efficient group signature schemes for large group. In: Proc. Of crypto'97, LNCS, Springer-Verlag, 410-424.

[16] Chen, C. L., Jan, J. K.(1999). A Novel Sealed-Bid Protocol in Networks, Proceedings of the 9th Information Security Conference, Taiwan, 1999, pp. 50-57.

[17] Juang,W.-S., Liaw, H.-T., Lin, P.-C., and Lin, C.-K. (2005). The Design of a Secure and Fair Sealed-bid Auction Service. Mathematical and Computer Modelling 41(8-9), pp. 973-985.

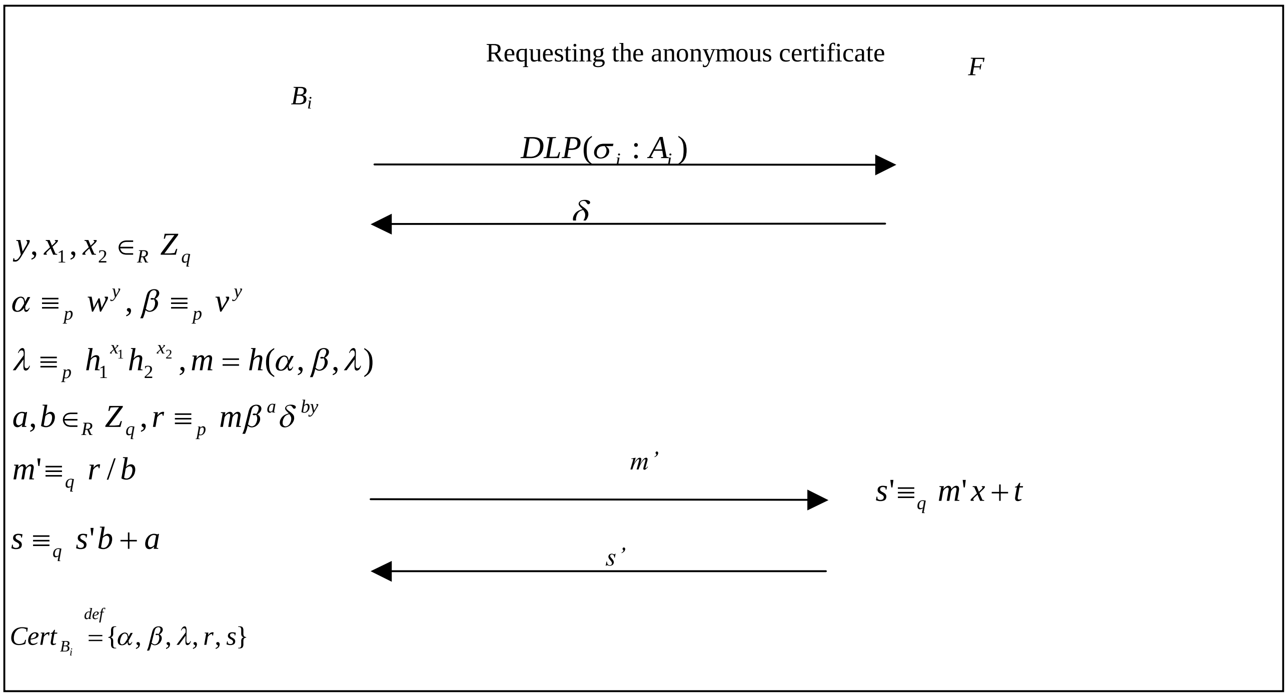
Requesting the anonymous certificate
B_i
F
$DLP(\sigma_i : A_i)$
δ
$y, x_1, x_2 \in_R Z_q$
$\alpha \equiv_p w^y, \beta \equiv_p v^y$
$\lambda \equiv_p h_1^{x_1} h_2^{x_2}, m = h(\alpha, \beta, \lambda)$
$a, b \in_R Z_q, r \equiv_p m\beta^a \delta^{by}$
$m' \equiv_q r / b$
m'
$s' \equiv_q m'x + t$
$s \equiv_q s'b + a$
s'
$Cert_{B_i} \overset{def}{=} \{\alpha, \beta, \lambda, r, s\}$

Figure 1. Requesting the anonymous certificate

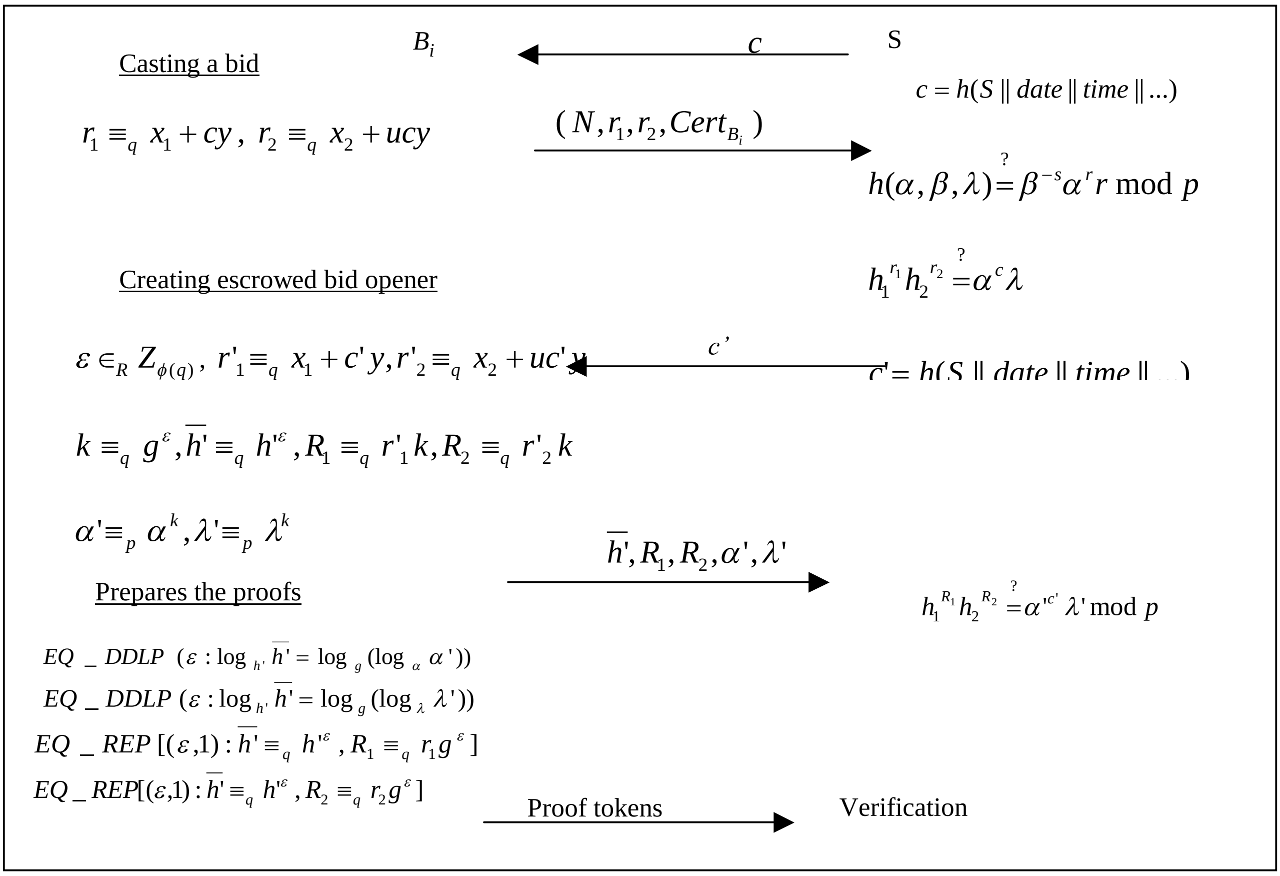
Casting a bid
B_i
c
S
$c = h(S \| date \| time \| ...)$
$r_1 \equiv_q x_1 + cy,\ r_2 \equiv_q x_2 + ucy$
$(N, r_1, r_2, Cert_{B_i})$
$h(\alpha, \beta, \lambda) \overset{?}{=} \beta^{-s} \alpha^r r \bmod p$
Creating escrowed bid opener
$h_1^{r_1} h_2^{r_2} \overset{?}{=} \alpha^c \lambda$
$\varepsilon \in_R Z_{\phi(q)},\ r'_1 \equiv_q x_1 + c'y, r'_2 \equiv_q x_2 + uc'y$
c'
$c' = h(S \| date \| time \| ...)$
$k \equiv_q g^{\varepsilon}, \overline{h'} \equiv_q h'^{\varepsilon}, R_1 \equiv_q r'_1 k, R_2 \equiv_q r'_2 k$
$\alpha' \equiv_p \alpha^k, \lambda' \equiv_p \lambda^k$
$\overline{h'}, R_1, R_2, \alpha', \lambda'$
Prepares the proofs
$h_1^{R_1} h_2^{R_2} \overset{?}{=} \alpha'^{c'} \lambda' \bmod p$
$EQ\_DDLP\ (\varepsilon : \log_{h'} \overline{h'} = \log_g (\log_\alpha \alpha'))$
$EQ\_DDLP\ (\varepsilon : \log_{h'} \overline{h'} = \log_g (\log_\lambda \lambda'))$
$EQ\_REP\ [(\varepsilon, 1) : \overline{h'} \equiv_q h'^{\varepsilon}, R_1 \equiv_q r_1 g^{\varepsilon}]$
$EQ\_REP[(\varepsilon, 1) : \overline{h'} \equiv_q h'^{\varepsilon}, R_2 \equiv_q r_2 g^{\varepsilon}]$
Proof tokens
Verification

Figure 2. bid-casting protocol

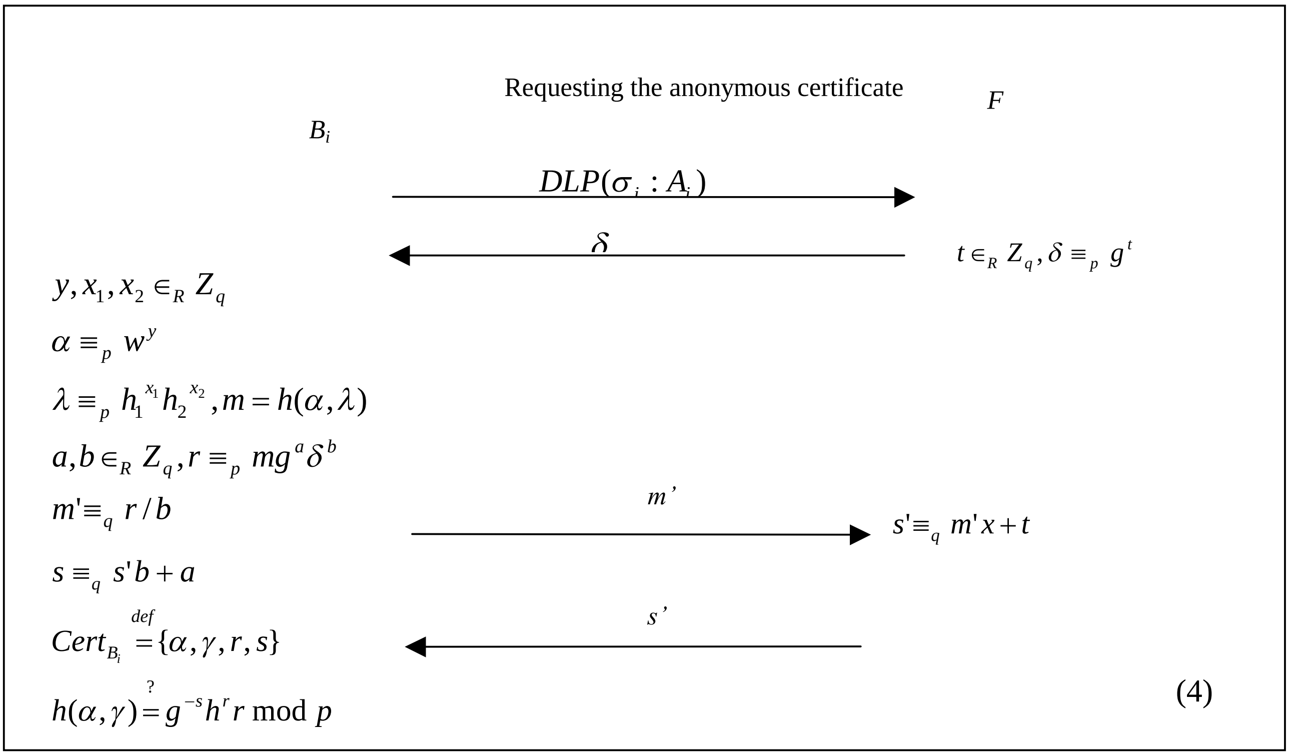
Requesting the anonymous certificate
B_i
F
$DLP(\sigma_i : A_i)$
δ
$t \in_R Z_q, \delta \equiv_p g^t$
$y, x_1, x_2 \in_R Z_q$
$\alpha \equiv_p w^y$
$\lambda \equiv_p h_1^{x_1} h_2^{x_2}, m = h(\alpha, \lambda)$
$a, b \in_R Z_q, r \equiv_p m g^a \delta^b$
$m' \equiv_q r / b$
m'
$s' \equiv_q m' x + t$
$s \equiv_q s' b + a$
s'
$Cert_{B_i} \overset{def}{=} \{\alpha, \gamma, r, s\}$
$h(\alpha, \gamma) \overset{?}{=} g^{-s} h^r r \bmod p$
(4)

Figure 3. Improved certificate generation phase8